UNCLASSIFIED DTIC FILE COPY

SECURITY CLASSIFICATION OF THIS PAGE

(2)

REPORT DOCUMENT

Form Approved OMB No. 0704-0188

1a. REPORT SECURITY CLASSIFICATION
UNCLASSIFIED

DTIC ELECTE APR 30 1990 S D

AD-A221 181

4. PERFORMING ORGANIZATION REPORT NUMBER(S)
PUIC MIQAP-J-006

5. MONITORING ORGANIZATION REPORT NUMBER(S)
PUIC MIQAP-J-006

6a. NAME OF PERFORMING ORGANIZATION
USAMICOM, Product Assurance Directorate

6b. OFFICE SYMBOL (If applicable)
AMSMI-QA-CF

7a NAME OF MONITORING ORGANIZATION
Product Assurance Directorate

6c. ADDRESS (City, State, and ZIP Code)
Redstone Arsenal, Al
Huntsville, Al 35898-5250

7b. ADDRESS (City, State, and ZIP Code)
Commander, USAMICOM, ATTN: AMSMI-QA
Redstone Arsenal, AL 35898-5290

8a. NAME OF FUNDING / SPONSORING ORGANIZATION
Product Assurance Directorate

8b. OFFICE SYMBOL (If applicable)
AMSMI-QA

9. PROCUREMENT INSTRUMENT IDENTIFICATION NUMBER
PB019-89

8c. ADDRESS (City, State, and ZIP Code)
Commander, USAMICOM
ATTN: AMSMI-QA
Redstone Arsenal, Al 35898-5290

10. SOURCE OF FUNDING NUMBERS

PROGRAM ELEMENT NO.	PROJECT NO.	TASK NO.	WORK UNIT ACCESSION NO.
PUIC	MIQAP	J	006

11. TITLE (Include Security Classification)
(U) Establish Deficiency Report Management System

12. PERSONAL AUTHOR(S)
Jim Marble

13a. TYPE OF REPORT	13b. TIME COVERED	14. DATE OF REPORT (Year, Month, Day)	15. PAGE COUNT
Final Technical	FROM 7/28/89 TO 9/30/89	89/9/28	66

16. SUPPLEMENTARY NOTATION
(U) Deficiency Report Management System

17. COSATI CODES		
FIELD	GROUP	SUB-GROUP

18. SUBJECT TERMS (Continue on reverse if necessary and identify by block number)
Developing Procedures for QDR Report Processing

19. ABSTRACT (Continue on reverse if necessary and identify by block number)
(U) Internal system for management control of deficiency report processing. Effort includes assessing activities, assigning management goals, measuring progress and periodic updating of procedures for efficient Quality Deficiency Reports (QDRs).

20. DISTRIBUTION / AVAILABILITY OF ABSTRACT
☒ UNCLASSIFIED/UNLIMITED ☐ SAME AS RPT. ☐ DTIC USERS

21. ABSTRACT SECURITY CLASSIFICATION
UNCLASSIFIED

22a. NAME OF RESPONSIBLE INDIVIDUAL
Janell Chaney

22b. TELEPHONE (Include Area Code)
205-876-5332

22c. OFFICE SYMBOL
AMSMI-QA-TI-PR

DD Form 1473, JUN 86 Previous editions are obsolete. SECURITY CLASSIFICATION OF THIS PAGE

HILTON SYSTEMS, INC. REPORT NO.: HSI/89-0013-04/1

ESTABLISHMENT OF DEFICIENCY REPORT MANAGEMENT SYSTEM

FINAL TECHNICAL REPORT

29 SEPTEMBER 1989

Prepared For: Commander
U.S. Army Missile Command
ATTN: Mr. Thomas L. Moore
Product Assurance Directorate
AMSMI-QA-CF
Redstone Arsenal, AL 35898-5290

The Contractor, Hilton Systems, Inc., hereby certifies that, to the best of its knowledge and belief, the technical data delivered herewith under Contract DAAH01-89-D-0013 is complete, accurate, and complies with all requirements of the contract.

PREPARED BY: M. Wayne Miller, Jr. **CONTRACT:** DAAH01-89-D-0013/0005

CLASSIFICATION AND CONTENT APPROVED BY: Harold R. Bright, Director, Product Assurance **DATA ITEM:** A003 (SOW Para 5.3)

HILTON SYSTEMS, INC.

29 September 1989 REPORT NO.: HSI/89-0013-04/1

Commander
U.S. Army Missile Command
ATTN: Mr. Thomas L. Moore
Product Assurance Directorate
AMSMI-QA-CF
Redstone Arsenal, AL 35898-5290

SUBJECT: **FINAL TECHNICAL REPORT**
CONTRACT NO. DAAH01-89-D-0013, DELIVERY ORDER 0005

Dear Mr. Moore:

Forwarded herewith are the subject report for the ESTABLISHMENT OF DEFICIENCY REPORT MANAGEMENT SYSTEM for the period of performance 07 March - 07 September 1989. The DD Form 250 will be forwarded with the next invoice.

This report does not contain any proprietary technical data which was developed independently by HILTON SYSTEMS, INC. Accordingly, there are no restrictions in the use of these data as defined by FAR 52.227-7013 dated 1 April 1988. The views, opinions and findings contained in this document are those of the author(s) and should not be construed as a position of the Product Assurance Directorate of the U.S. Army Missile Command.

It has been a pleasure to perform this work for the Product Assurance Directorate. If you have any questions regarding the enclosed, please do not hesitate to call the undersigned at 883-2260.

Sincerely,

Harold R. Bright
Operations Manager
Product Assurance Division

Enclosure

HRB/km

cc: DCASMA/Birmingham
AMSMI-QA-TI-PB/Ms. Janell Chaney (cover letter only)

2227 DRAKE AVENUE, S.W. • SUITE 31 • HUNTSVILLE, ALABAMA 35805 • (205) 883-2260 • FAX (205) 883-7062
CORPORATE OFFICE • 920 FARMER STREET • VICKSBURG, MISSISSIPPI 39180 • (601) 634-6164 • FAX (601) 638-2342

ABSTRACT

The Customer Feedback Office (CFO) of the Product Assurance Directorate (PAD), MICOM, has the mission to manage and analyze data in the Deficiency Reporting System and make recommendations for appropriate actions. This report presents the results of a contract delivery order performed by the Product Assurance Division of Hilton Systems, Inc., for support to the CFO.

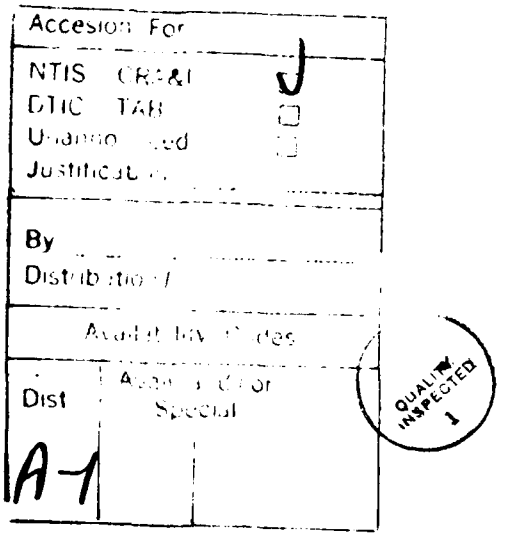


TABLE OF CONTENTS

1.0 Introduction	1
2.0 Discussion	2
2.1 Review Empress/32	2
2.2 Investigate Graphics Capabilities	2
2.3 Match S2K to Empress/32	3
2.4 Prepare Prototype Summary Report	3
2.5 Prepare Sort Routine	30
2.6 Prepare Outgoing Letters	30
3.0 Status of Accomplishments	45
4.0 Summary	50
5.0 Conclusion	50
6.0 Recommendations	50
APPENDIX A - Task Plan and Milestone Chart, 17 March 1989	A-1
APPENDIX B - Main Menu Source Code	B-1
APPENDIX C - Empress 32/ASCII Matrix File Menu Source Code	C-1
APPENDIX D - Weapon System QDR Summary Source Code	D-1
APPENDIX E - Action Officer QDR Summary Source Code	E-1
APPENDIX F - Contractor QDR Summary Source Code	F-1
APPENDIX G - QDR Status Report Source Code	G-1
APPENDIX H - QDR Status Barchart Menu Source Code	H-1
APPENDIX I - Special Functions Menu Source Code	I-1
APPENDIX J - Reorder Menu Source Code	J-1
APPENDIX K - Memo Utility for DRS Update	K-1
APPENDIX L - QDR Alerting System	L-1
APPENDIX M - Laser Printer/Font Program	M-1

1.0 Introduction

The contractor was tasked by Delivery Order (DO) 0005 to establish a Product Assurance Directorate (PAD) internal system for management control of deficiency report processing activities including Quality Deficiency Reports (QDRs), Equipment Improvement Reports (EIRs), and Warranty Claim Actions (WCAs). Delivery Order 020 was issued as an expansion of DO #5 for two special tasks. This report includes those portions of DO #20 that pertain to work added to DO #5. This final technical report discusses the contractor's actions taken to overcome the difficulties encountered in dealing with the lack of valid data being provided to the Customer Feedback Office, prior to 11 August, 1989, and accomplishments after that date.

Progress made in meeting the requirements of the SOWs is detailed herein, and this report follows the format of data item description, UDI-S-23272C. The discussion relates to the approved task plan attached as Appendix A.

2.0 Discussion

2.1 Review Empress/32

Empress/32 is the database management program that was purchased by the Government as part of the Unisys 5000 computer package. Technical support for this program was to be provided through the Information Management Directorate (IMD); thus, the authority to talk directly with a technical representative of Empress was not available to the contractor. This posed a problem in that the available Empress documentation was considered lacking in many areas and much time was spent in trial and error efforts to get functions to perform properly.

A major problem encountered was that the Empress software (as installed) would not link user-developed software using host languages (FORTRAN, C, PASCAL). A solution to this problem is scheduled for early October 1989.

In spite of this, Empress is a viable database manager which could be made to perform as needed if direct access to Empress technical representatives was available.

2.2 Investigate Graphics Capabilities

The Customer Feedback Office has an on-going requirement for viewgraph capability for presentation of database information at all levels of command. They have available a Macintosh SE computer with Excel software and a Unisys PC with Excel and Lotus programs, and Harvard Graphics is on order. There is a chart program in the Unisys office manager system (OFIS) that produces standard business charts (e.g., pie, line, bar charts). This is used in conjunction with a Sweet Pea six pin color plotter for making viewgraphs.

The Deficiency Reporting System (DRS) database information is loaded on the Customer Feedback Office Unisys 5000 from IMD's IBM mainframe in the form of 21 ASCII text files. These System 2000 (S2K) generated text files represent database tables of various field lengths to be imported into the Empress database management system (DBMS). The integration of AWK programs, MSbatch routines, and Unix shell scripts provides the ability to manipulate and prepare data for producing the required CFO graphics. These shell scripts will also enable the task requirements to be performed from the Unix operating system level instead of being limited to operations within the database management system.

The methodology for producing the CFO Weapon System QDR status charts consists of the following steps:

1. Create an ASCII text file containing the relevant data from the DRS database.
2. Use the Kermit file transfer protocol to download the ASCII file to a Macintosh SE.
3. Place the downloaded ASCII file into an Excel template file.
4. Prepare the Excel template file to produce the charts.
5. Use a laser printer to print the desired charts.

The Excel template file contains the necessary logic to produce specific QDR status charts in terms of their numerical values and presentational qualities.

2.3 Match S2K to Empress/32

2.3.1 A couple of major problems were encountered in this task, the largest being that the original process for downloading S2K data required dates entered in month-day-year sequence, while Empress requires a year-month-day sequence. A request was made to IMD early in the task to correct this problem, and it was solved as of 11 August 1989, 5 months into the task.

Prior to this date the contractor was able to develop an executable Unix shell script incorporating several AWK programs to correct the date problem. To execute the program, a list of parameters are passed to the shell script relating to the specific table being corrected. The resulting output consists of a new table with the reordered date field in the proper format. This routine has been placed in the selection menu in the event it may be needed for future applications. If needed, it is recommended that it be run at night because for large files it slows the CPU considerably.

2.3.2 Null fields cannot be entered from S2K to Empress. Prior to the 11 August solution noted above, the contractor entered common characters in the null fields to get the data into Empress and then subsequently deleted them.

2.3.3 Having 21 tables makes query almost impossible time-wise, since all the tables that contained needed data elements had to be joined together before the data could be retrieved. An attempt was made to combine these tables into one large table. Problems were encountered in the Unisys with inherent file size limitations. This is still under investigation because it could cause serious problems for CFO as they add needed data fields to the database.

The latest download from IMD consists of 12 tables following the format of the S2K DRS database and is considered adequate for the tasks in this delivery order.

2.3.4 While the original S2K data download contained some erroneous data, the contractor used good portions of the IMD downloaded data and built a strawman Empress database. This allowed for the programming and testing of application software. As stated earlier, a good database was loaded on 11 August 1989, and a demo of menu and application software was made on 17 August 1989. User comments and desires were noted and incorporated.

2.4 Prepare Summary Report

This task requires that an automated QDR processing system be constructed to allow CFO personnel to more efficiently analyze and output QDR information from the DRS database. The following paragraphs discuss the interactive menu-driven system and the summary reports developed by the contractor for CFO.

2.4.1 Main Menu

The QDR information is accessed and manipulated by the contractor-developed interactive menu-driven program "xcfmenu." Upon entering "xcfmenu," the main menu (Figure 1) appears on the screen.

Features of the menu include: day, date and time that the menu program is run and single letter code entry for making a task selection. If an invalid code is entered, the message, "You Have Made An Invalid Selection! Please Try Again," will appear and, after a short delay, the menu will reappear.

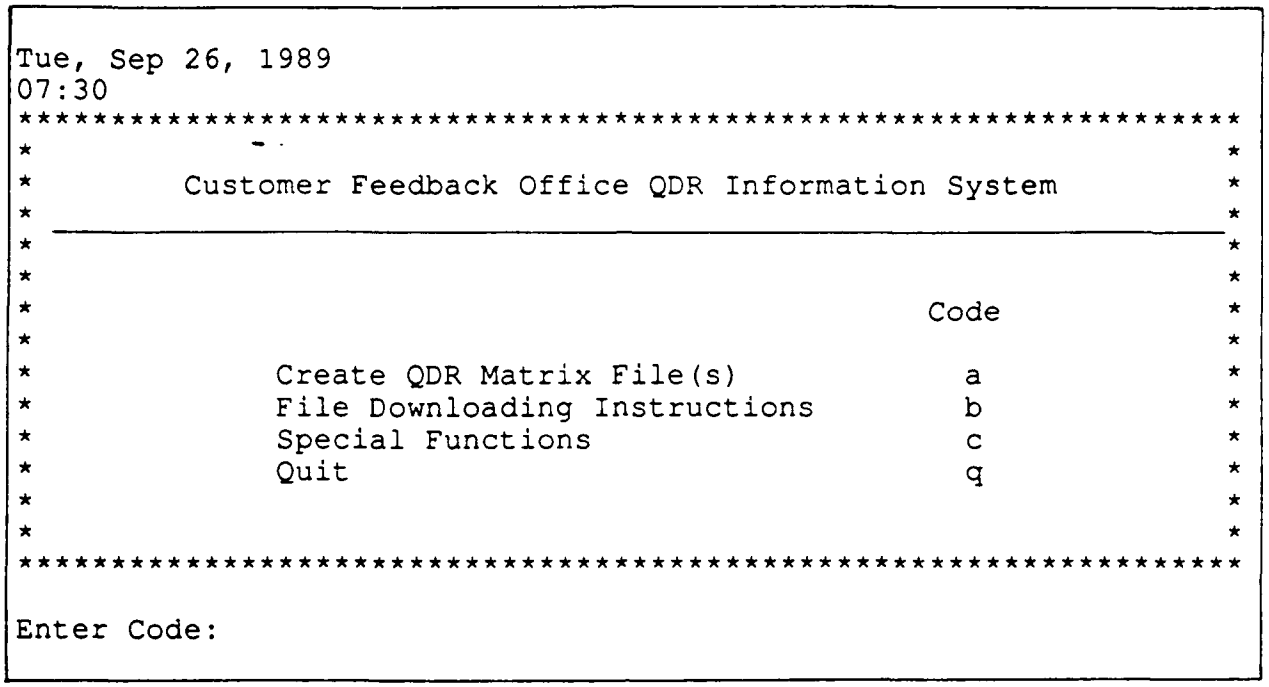


Figure 1

The contractor-developed source code for this menu-driven program is attached as Appendix B. A look at this code will reveal that it is comprised of a number of modules each consisting of a customized set of functions. Each module performs a specific function that allows the QDR data to be processed according to CFO requirements.

2.4.2 Empress 32/ASCII Matrix File Menu

The selection of code "a" on the Main Menu will present the user with the Empress 32/ASCII Matrix File Menu (Figure 2). By entering one of the numbered choices, this menu allows the user to select one of four interactive reports or return to the Main Menu.

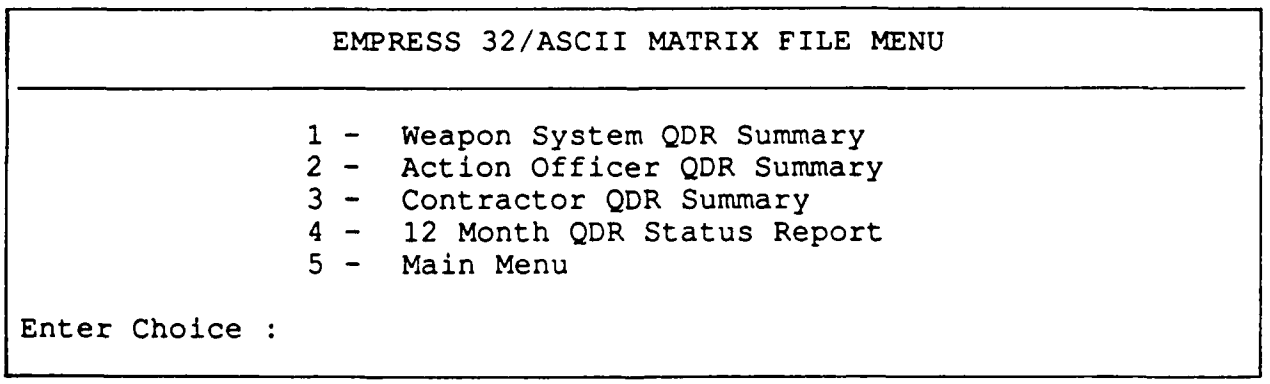


Figure 2

The source code (Appendix C) for this process consists of a resident set of SQL expressions that will return a matrix, in ASCII format, summarizing the QDR status for a given month. The QDR milestones summarized are listed in Table 2-1.

QDR Milestones

Open Beginning of Period (OBOP)
Received During Period (RCVD)
Closed During Period (CLOSED)
Transferred During Period (TRANSF)
Open End of Period (OEOP)
1 - 119 Days
120 - 180 Days
Over 180 Days
Over 180 Days Closed During Period

Table 2-1

2.4.3 Weapon System QDR Summary

Selection of the number "1" on the Empress 32/ASCII Matrix File Menu activates the source code (Appendix D) necessary to produce the Quality Deficiency Report Summary by weapon system (Figure 3).

The program starts with the screen presentation shown in Figure 4. Each line on this screen is prompted individually (i.e., as one prompt is satisfied, the next one will appear). A filename is requested as a way for the user to archive Figure 3.

QUALITY DEFICIENCY REPORT SUMMARY
AMSMI-QA-CF

SUMMARY PERIOD : AUGUST 1989
(as of 22 Sep 1989)

WEAPON SYSTEM	OBOP	RCVD	CLOSED	TRANSF	OEOP	1-119 DAYS	120-180 DAYS	OVER 180	OVR 180 CLOSED
OH-58D	10	3	0	1	12	4	8	0	0
CHAPARRAL	91	3	6	3	90	16	24	50	1
HAWK	65	5	17	0	53	20	11	22	3
TARGETS	20	7	2	0	25	14	8	3	0
REDEYE	5	0	1	0	4	0	1	3	1
STINGER	16	2	1	3	14	8	4	2	0
PATRIOT	34	14	19	1	28	20	4	4	3
FAAR	11	0	2	0	9	3	5	1	0
MLRS	65	4	9	0	61	11	13	37	7
LANCE	5	3	3	2	3	3	0	0	3
HELLFIRE	11	3	6	0	8	5	3	0	3
LCSS	0	1	0	0	1	1	0	0	0
TOW	64	9	16	0	57	18	12	27	10
PERSHING II	19	0	0	0	19	3	5	11	0
DRAGON	18	3	3	1	17	17	0	0	1
TOW BRADLEY	26	11	1	0	36	22	9	5	0
GLLD	0	0	0	0	0	0	0	0	0
MULE	0	0	0	0	0	0	0	0	0
THERM IMAGE	15	2	2	0	15	13	1	1	2
TADS	12	11	0	0	23	21	2	0	0
CALIBRATION	3	0	1	0	2	0	2	0	0
APACHE	2	0	0	8	2	1	0	1	0
AN/TSQ-73	1	0	1	0	0	0	0	0	0
TOW COBRA	21	10	6	0	25	13	4	8	2
FIST-V	1	2	0	0	3	3	0	0	0
UNKNOWN	0	2	0	1	2	2	0	0	0
TOTAL	515	95	96	20	509	218	116	175	36

FIGURE 3

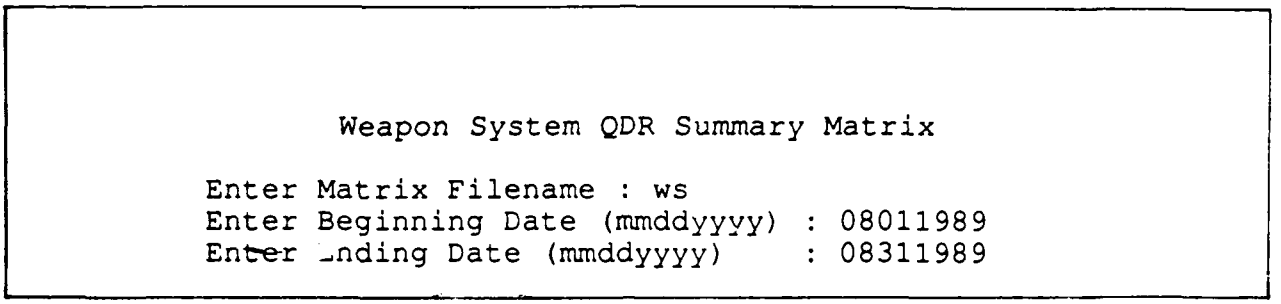


Figure 4

On this menu and on any of the following menus in which the user is to enter data, the entering of a "q" at the first prompt will return the user to the Main Menu.

Upon successfully satisfying the Empress SQL logic conditions, the weapon system Unix program performs a tally, by weapon system, of the number of QDRs against each weapon system for a given month. These QDRs are then stored in a temporary file for each respective QDR milestone. After all QDR milestone files are created, a matrix is formed by merging all nine milestone files together. Finally, the matrix is linked with an ASCII heading file relevant to the weapon system data. The weapon systems used for this analysis are listed in Table 2-2.

DRS Weapon Systems

OH-58D	Chaparral
Hawk	Targets
Redeye	Stinger
Patriot	FAAR
MLRS	Lance
Hellfire	LCSS
TOW	Pershing II
Dragon	TOW Bradley
GLLD	Mule
Thermal Imagery	TADS
Calibration	Apache
AN/TSQ-73	TOW Cobra
FIST-V	Unknown

Table 2-2

While the process is running, the message "Processing Weapon System QDR Data, Please Wait...." will appear with the start time in the upper left hand corner of the screen. At the conclusion of the run another message providing start time, user ID, group ID, file size in bytes, date, ending time and filename will appear.

```
09:53    rhaddox  hsi        1491 Sep 27 10:05 ws
```

Messages such as these will appear for all programs selected from the Empress 32/ASCII Matrix File Menu.

2.4.4 Action Officer QDR Summary

The action officer Unix program (Appendix E) performs the same function as the weapon system program except for the replacement of the weapon system variable with action officer (Figure 5). The program is initiated by selecting number "2" on the Empress 32/ASCII Matrix File Menu. The same prompt procedures are presented (Figure 6) as with "weapon system."

```
        Action Officer QDR Summary Matrix

Enter Matrix Filename : ao
Enter Beginning Data (mmddyyyy) : 08011989
Enter Ending Date (mmddyyyy)    : 08311989
```

Figure 6

2.4.5 Contractor QDR Summary

The contractor Unix program (Appendix F) executes the same procedures as weapon system and action officer except for the heading material. This program was approached differently since an analysis of the contractor data yielded 328 unique contractors. Due to this occurrence, the tally files for each QDR milestone were not merged together to form a matrix as shown for weapon system, etc. As a result, each of the nine contractor QDR milestone files is linked, with separate identification headings, and printed separately (Figure 7).

This program is initiated by selecting number "3" on the Empress 32/ASCII Matrix File Menu which causes the prompts shown in Figure 8 to appear.

```
        Contractor QDR Summary Matrix

Enter Matrix Filename : cntr
Enter Beginning Date (mmddyyyy) : 08011989
Enter Ending Date (mmddyyyy)    : 08311989
```

Figure 8

QUALITY DEFICIENCY REPORT SUMMARY
AMSMI-QA-CF

SUMMARY PERIOD : __AUGUST__1989_____
(as of 22 Sep 1989)

ACTION OFFICER	OBOP	RCVD	CLOSED	TRANSF	OEOP	1-119 DAYS	120-180 DAYS	OVER 180	OVR 180 CLOSED
L. Blanton	173	25	28	9	171	53	40	78	19
G. Ross	127	14	43	0	98	41	27	30	11
S. Scott	37	11	7	3	38	28	10	0	4
J. Cooper	132	1	10	4	124	18	39	67	2
B. James	17	17	3	1	30	30	0	0	0
R. Thomas	7	4	1	0	10	10	0	0	0
E. Ashworth	10	2	0	0	12	12	0	0	0
M. Jones	12	18	4	0	26	26	0	0	0
Cust Fdbk Ofc	0	4	0	4	0	0	0	0	0
TOTAL	515	96	96	21	509	218	116	175	36

FIGURE 5

CONTRACTOR - QUALITY DEFICIENCY REPORT SUMMARY
AMSMI-QA-CF

SUMMARY PERIOD : AUGUST 1989
(as of 22 Sep 1989)

CNTR CODE	CONTRACTOR	OBOP
00641	FORWAY INDUSTRIES INC	2
01178	LOURDES INDUSTRIES, INC.	1
01351	SOUTHERN GEAR & MACHINE	1
01848	TARTAN INDUSTRIES	1
02731	HUGHES HELICOPTERS	1
03104	TELEDYNE CAE	1
04939	MARTIN MARIETTA AEROSPACE CORP	11
05452	RAYTHEON CO MSL SYS DIV	1
05716	RAYTHEON CO	17
06509	ENGINEERED MAGNETICS INC.	1
07145	TIMES FIBER COMMUNICATIONS CO.	1
07187	SPERRY CORP	8
07239	BIDDLE INSTRUMENTS	1
09087	INFODEX, INC.	1
0A4Y9	DBA SYSTEMS INC	3
0AH14	DEFENSE TECHNOLOGIES INC.	1
11871	CAI DIVISION OF RECON OPTICAL	1
13310	AEROJET SOLID PROPULSION CO.	1
14339	GENERAL RELIANCE CORP	1
15090	HUGHES AIRCRAFT CO	4
15984	MILAN BOX CORP	3
16004	DAVEY COMPRESSOR CO	2
17029	A C INC.	2
18355	MCDONNELL DOUGLAS CORP	1
19605	INSCOM ELECTRONICS CORP	4
19710	MPC PRODUCTS CORP	1
19976	UNICOR	3
1A228	ABSTON MACHINE SHOP INC	2
1BG56	COMMAND TECHNOLOGY - SALT LAKE INC.	1
1CG31	J & J PRECISION MACH SHOP	1
1EV35	DENHAM MACHINE & TOOL	1
1HE79	RAYTHEON	2
1L007	SPECIAL PROJ MACH & TOOL	2
20418	EMERSON ELEC CO	2
22255	LINDGREEN R F ENCLOSURES INC	2
22830	SYMETRICS INDUSTRIES INC.	1
23302	S W ELECTRONICS & MFG CO	2
24321	CONDOR PACIFIC INDUSTRIES INC	1
26116	AIRCON FILTER MFG CO INC.	2
26530	PBR ELECTRONICS INC	17
28861	MCDONNELL DOUGLAS MSL SYSTEMS CORP	3
29055	TRAM ELECTRONICS INC	2
30670	G AND D AIRCRAFT PARTS INC.	1
32067	MARVIN ENGINEERING	2
32206	SAVE-O-SEAL	1
34449	B-K MFG CO	9
34705	CLIFFDALE MFG. CO.	1
3L775	UNIVERSAL TECHNOLOGIES	2

FIGURE 7

3V055	ANCO MACHINE CO.	2
3Y012	SPACECRAFT MACHINE CO	7
43715	TRONTECH INC.	1
48294	PSI PERIPHERAL SUPPORT	1
4K245	SMALL BUSINESS ADMINISTRATION	2
4M241	BRAMBLETT ELECTRONICS CO. INC.	1
4X740	PREFERRED METALS	1
51215	ROCKWELL INTERNATIONAL CORP.	1
51753	ARISTA DEVICES CORP	3
52202	MODULAR DEVICES INC	4
53753	UNION FED CORR INSTITUTION	3
54736	UNICOR	3
55096	FRASER VOLPE CORPORATION	2
55106	VALLEY ENTERPRISE INC.	3
55452	PACER SYSTEMS	2
55928	FEDERAL PRISONS INDUSTRIES	2
56132	REYNOLDS AND TAYLOR INC.	1
56853	UNICOR FEDERAL PRISON	2
58163	NOAH HOWDEN INC.	1
58339	TARTAN INDSTRIES	3
5K815	ANIXTER TULSA INC.	1
5U566	JOMA MFG INC.	1
61453	R.S. MICROWAVE CO INC.	4
62313	VOUGHT CORP MLRS DIV	3
62445	DEUTZ CORP	4
62502	GOODMAN BALL, INC.	1
62810	APACHE ENTERPRISES INC	1
62983	VICKERS INC.	2
63911	MILITARY AIRCRAFT SALES INC	2
64031	ARRAL INDUSTRIES	1
64059	LTV AEROSPACE & DEF CO	9
64143	BESAFA INC	1
64448	R & D MACHINE CO.	7
64609	BANNER MACHINE CO.	1
66015	HDL RESEARCH LAB INC	1
6K329	U.S. SMALL BUSINESS ADMIN	4
6X173	WILDWOOD ELECTRONICS INC	4
70898	BEECH AIRCRAFT CO	7
7D457	RIDGE INSTRUMENTS CO INC.	1
7M676	PRECISION MACHINING INC	1
7N710	CONSOLIDATED INDUSTRIES INC.	3
7T482	MILITARY AIRCRAFT SALES CO	1
80212	FMC CORP	1
80293	FLUIDS CONTROL	1
80378	VOUGHT CORPORATION	35
81579	MASON ELECTRIC CO INC	1
81902	CRAIG SYSTEMS CORP	1
82577	HUGHES AIRCRAFT CO	47
82686	TRANSICOIL, INC.	1
89536	FLUKE JOHN MFG CO	1
8A549	PRECISION SPECIALTY CORP	1
8T985	SWAIM MACHINE CO	1
91196	FISHER ENGINEERING INC	1
92059	RAYMOND ENGINEERING INC	5
92837	UNICOR	8
94580	HONEYWELL INC	2
95542	NORDEN SYSTEMS	2
96214	TEXAS INSTRUMENTS	15
96906	MILITARY STANDARDS PROMULGATED	1
97928	DEUTSCH FASTENER CORP	1

98255	SOUTHWEST MOBILE SYSTEMS CORP	1
98284	HUMPHRY INC	1
98453	NORTHROP CORP	2
98659	COMPUTER INSTRUMENTS CORP	3
98889	THE TALLEY CORPORATION	1
99251	LITTON PRECISION PRODUCTS	1
99313	VARIAN ASSOCIATES INC	1
99584	GENERAL DYNAMICS POMONA DIVISION	4
9E579	HEWLETT PACKARD COMPANY	1
9N498	LAND L PRECISION MACH CO	1
9S515	DELTA LIGHTING CORP.	1
9X991	TRICELIN CORP.	1
9Y279	RPL ELECTRONICS CO	2
XXXXX	AMOS METAL PRODUCTS	1

CONTRACTOR - QUALITY DEFICIENCY REPORT SUMMARY
AMSMI-QA-CF

SUMMARY PERIOD : AUGUST 1989
(as of 22 Sep 1989)

CNTR CODE	CONTRACTOR	RCVD
05716	RAYTHEON CO	5
07187	HONEYWELL INC	2
17863	TELEDYNE SYSTEMS CO	1
29055	TRAM ELECTRONICS INC	1
40968	ISLAND COMPONENTS INC	1
55452	PACER SYSTEMS	1
55973	ALLIED BENDIX AEROSPACE	1
56853	UNICOR	1
80378	VOUGHT CORPORATION	2
82577	HUGHES AIRCRAFT CO	20
92837	UNICOR	1
95542	NORDEN SYSTEMS INC	1
9S515	DELTA LIGHTING CORP	1

CONTRACTOR - QUALITY DEFICIENCY REPORT SUMMARY
AMSMI-QA-CF

SUMMARY PERIOD : AUGUST 1989
(as of 22 Sep 1989)

CNTR CODE	CONTRACTOR	CLOSED
00641	FORWAY INDUSTRIES INC	2
01848	TARTAN INDUSTRIES	1
04939	MARTIN MARIETTA AEROSPACE CORP	3
05716	RAYTHEON CO	2
0AH14	DEFENSE TECHNOLOGIES INC.	1
13310	AEROJET SOLID PROPULSION CO.	1
15984	MILAN BOX CORP	1
17029	A C INC.	1
19976	UNICOR	1
1A228	ABSTON MACHINE SHOP INC	2
1BG56	COMMAND TECHNOLOGY - SALT LAKE INC.	1
1CG31	J & J PRECISION MACH SHOP	1
20418	EMERSON ELEC CO	1
26116	AIRCON FILTER MFG CO INC.	2
26530	PBR ELECTRONICS INC	2
34449	B-K MFG CO	4
3L775	UNIVERSAL TECHNOLOGIES	1
4K245	SMALL BUSINESS ADMINISTRATION	2
51215	ROCKWELL INTERNATIONAL CORP.	1
52202	MODULAR DEVICES INC	3
53753	UNION FEDERAL CORR INSTITUTION	1
55928	FEDERAL PRISONS INDUSTRIES	1
56853	UNICOR FEDERAL PRISONS INDUS	1
63911	MILITARY AIRCRAFT SALES INC	2
64448	R & D MACHINE CO.	7
6X173	WILDWOOD ELECTRONICS INC	1
7M676	PRECISION MACHINING INC	1
7T482	MILITARY AIRCRAFT SALES CO	1
80293	FLUIDS CONTROL	1
80378	VOUGHT CORPORATION	2
81039	PLESSEY DYNAMICS CORP	1
81579	MASON ELECTRIC CO INC	1
81902	CRAIG SYSTEMS CORP	1
82577	HUGHES AIRCRAFT CO	6
89536	FLUKE JOHN MFG CO	1
8A549	PRECISION SPECIALTY CORP	1
92837	UNICOR FED CORR INSTITUTION	1
96214	TEXAS INSTRUMENTS CORP.	2
98255	SOUTHWEST MOBILE SYSTEMS CORP	1
99251	LITTON PRECISION PRODUCTS	1
99313	VARIAN ASSOCIATES INC	1
9Y279	RPL ELECTRONICS CO	2

CONTRACTOR - QUALITY DEFICIENCY REPORT SUMMARY
AMSMI-QA-CF

SUMMARY PERIOD : __AUGUST__1989_____
(as of 22 Sep 1989)

CNTR CODE	CONTRACTOR	TRANSF
29055	TRAM ELECTRONICS INC	1
80212	FMC CORP	1

CONTRACTOR - QUALITY DEFICIENCY REPORT SUMMARY
AMSMI-QA-CF

SUMMARY PERIOD : __AUGUST__1989_____
(as of 22 Sep 1989)

CNTR CODE	CONTRACTOR	OEOP
01178	LOURDES INDUSTRIES, INC.	1
01351	SOUTHERN GEAR & MACHINE	1
02731	HUGHES HELICOPTERS	1
03104	TELEDYNE CAE	1
04939	MARTIN MARIETTA AEROSPACE CORP	8
05452	RAYTHEON CO MSL SYS DIV	1
05716	RAYTHEON CO	20
06509	ENGINEERED MAGNETICS INC.	1
07145	TIMES FIBER COMMUNICATIONS CO.	1
07187	HONEYWELL INC	10
07239	BIDDLE INSTRUMENTS	1
09087	INFODEX, INC.	1
0A4Y9	DBA SYSTEMS INC	3
11871	CAI DIVISION OF RECON OPTICAL	1
14339	GENERAL RELIANCE CORP	1
15090	HUGHES AIRCRAFT CO	4
15984	MILAN BOX CORP	2
16004	DAVEY COMPRESSOR CO	2
17029	A C INC.	1
17863	TELEDYNE SYSTEMS CO	1
18355	MCDONNELL DOUGLAS CORP	1
19605	INSCOM ELECTRONICS CORP	4
19710	MPC PRODUCTS CORP	1
19976	UNICOR FEDERAL PRISON	2
1EV35	DENHAM MACHINE & TOOL	1
1HE79	RAYTHEON	2
1L007	SPECIAL PROJ MACH & TOOL	2
20418	EMERSON ELEC CO	1
22255	LINDGREEN R F ENCLOSURES INC	2
22830	SYMETRICS INDUSTRIES INC.	1
23302	S W ELECTRONICS & MFG CO	2
24321	CONDOR PACIFIC INDUSTRIES INC	1
26530	PBR ELECTRONICS INC	15
28861	MCDONNELL DOUGLAS MSL SYSTEMS CORP	3
29055	TRAM ELECTRONICS INC	2
30670	G AND D AIRCRAFT PARTS INC.	1
32067	MARVIN ENGINEERING	2
32206	SAVE-O-SEAL	1
34449	B-K MFG CO.	5
34705	CLIFFDALE MFG. CO.	1
3L775	UNIVERSAL TECHNOLOGIES	1
3V055	ANCO MACHINE CO.	2
3Y012	SPACECRAFT MACHINE CO	7
40968	ISLAND COMPONENTS INC	1
43715	TRONTECH INC.	1
48294	PSI PERIPHERAL SUPPORT	1
4M241	BRAMBLETT ELECTRONICS CO. INC.	1
4X740	PREFERRED METALS	1

51753	ARISTA DEVICES CORP	3
52202	MODULAR DEVICES INC	1
53753	UNION FED CORR INSTITUTION	2
54736	UNICOR	3
55096	FRASER VOLPE CORPORATION	2
55106	VALLEY ENTERPRISE INC.	3
55452	PACER SYSTEMS	3
55928	FEDERAL PRISONS INDUSTRIES	1
55973	ALLIED BENDIX AEROSPACE	1
56132	REYNOLDS AND TAYLOR INC.	1
56853	UNICOR	2
58163	NOAH HOWDEN INC.	1
58339	TARTAN INDSTRIES	3
5K815	ANIXTER TULSA INC.	1
5U566	JOMA MFG INC.	1
61453	R.S. MICROWAVE CO INC.	4
62313	VOUGHT CORP MLRS DIV	3
62445	DEUTZ CORP	4
62502	GOODMAN BALL, INC.	1
62810	APACHE ENTERPRISES INC	1
62983	VICKERS INC.	2
64031	ARRAL INDUSTRIES	1
64059	LTV AEROSPACE & DEF CO	9
64143	BESAFA INC	1
64609	BANNER MACHINE CO.	1
66015	HDL RESEARCH LAB INC	1
6K329	U.S. SMALL BUSINESS ADMIN	4
6X173	WILDWOOD ELECTRONICS INC.	3
70898	BEECH AIRCRAFT CO	7
7D457	RIDGE INSTRUMENTS CO INC.	1
7N710	CONSOLIDATED INDUSTRIES INC.	3
80378	VOUGHT CORPORATION	35
82577	HUGHES AIRCRAFT CO	61
82686	TRANSICOIL, INC.	1
8T985	SWAIM MACHINE CO	1
91196	FISHER ENGINEERING INC	1
92059	RAYMOND ENGINEERING INC	5
92837	UNICOR	8
94580	HONEYWELL INC	2
95542	NORDEN SYSTEMS	3
96214	TEXAS INSTRUMENTS	14
96906	MILITARY STANDARDS PROMULGATED	1
97928	DEUTSCH FASTENER CORP	1
98284	HUMPHRY INC	1
98453	NORTHROP CORP	2
98659	COMPUTER INSTRUMENTS CORP	3
98889	THE TALLEY CORPORATION	1
99584	GENERAL DYNAMICS POMONA DIVISION	4
9E579	HEWLETT PACKARD COMPANY	1
9N498	LAND L PRECISION MACH CO	1
9S515	DELTA LIGHTING CORP	2
9X991	TRICELIN CORP.	1
XXXXX	AMOS METAL PRODUCTS	1

CONTRACTOR - QUALITY DEFICIENCY REPORT SUMMARY
AMSMI-QA-CF

SUMMARY PERIOD : AUGUST 1989
(as of 22 Sep 1989)

CNTR CODE	CONTRACTOR	1-119 DAYS
01351	SOUTHERN GEAR & MACHINE	1
05716	RAYTHEON CO	9
06509	ENGINEERED MAGNETICS INC.	1
07145	TIMES FIBER COMMUNICATIONS CO.	1
07187	HONEYWELL INC	3
0A4Y9	DBA SYSTEMS INC	3
15090	HUGHES AIRCRAFT CO.	1
15984	MILAN BOX CORP	1
16004	DAVEY COMPRESSOR CO	2
17863	TELEDYNE SYSTEMS CO	1
18355	MCDONNELL DOUGLAS CORP	1
19976	UNICOR FEDERAL PRISON	2
20418	EMERSON ELEC CO	1
22255	LINDGREEN R F ENCLOSURES INC	1
26530	PBR ELECTRONICS INC	1
28861	MCDONNELL DOUGLAS MSL SYSTEMS CORP	3
30670	G AND D AIRCRAFT PARTS INC.	1
34449	B-K MFG. CO.	3
3Y012	SPACECRAFT MACHINE CO.	1
40968	ISLAND COMPONENTS INC	1
43715	TRONTECH INC.	1
55096	FRASER VOLPE CORPORATION	2
55106	VALLEY ENTERPRISE INC.	3
55452	PACER SYSTEMS	3
55973	ALLIED BENDIX AEROSPACE	1
56853	UNICOR	2
58339	TARTAN INDSTRIES	3
5K815	ANIXTER TULSA INC.	1
5U566	JOMA MFG INC.	1
61453	R.S. MICROWAVE CO INC.	3
62983	VICKERS INC.	1
64059	LTV AEROSPACE AND DEFENSE CO.	1
70898	BEECH AIRCRAFT CORP.	1
7D457	RIDGE INSTRUMENTS CO INC.	1
80378	VOUGHT CORPORATION	3
82577	HUGHES AIRCRAFT CO	34
82686	TRANSICOIL, INC.	1
92837	UNICOR	1
95542	NORDEN SYSTEMS INC	1
96214	TEXAS INSTRUMENTS CORP.	5
99584	GENERAL DYNAMICS POMONA DIVISION	4
9E579	HEWLETT PACKARD COMPANY	1
9S515	DELTA LIGHTING CORP	2
9X991	TRICELIN CORP.	1

CONTRACTOR - QUALITY DEFICIENCY REPORT SUMMARY
AMSMI-QA-CF

SUMMARY PERIOD : AUGUST 1989
(as of 22 Sep 1989)

CNTR CODE	CONTRACTOR	120-180 DAYS
02731	HUGHES HELICOPTERS	1
04939	MARTIN MARIETTA AEROSPACE CORP	1
05452	RAYTHEON CO MSL SYS DIV	1
05716	RAYTHEON CO	2
07187	SPERRY CORP	7
07239	BIDDLE INSTRUMENTS	1
11871	CAI DIVISION OF RECON OPTICAL	1
15984	MILAN BOX CORP	1
19605	INSCOM ELECTRONICS CORP	1
1HE79	RAYTHEON CO MSL SYS DIV	1
1L007	SPECIAL PROJ MACH & TOOL	1
23302	S W ELECTRONICS & MFG CO	1
26530	PBR ELECTRONICS INC	10
29055	TRAM ELECTRONICS INC	1
32067	MARVIN ENGINEERING	2
34449	B-K MFG. CO.	1
34705	CLIFFDALE MFG. CO.	1
3Y012	SPACECRAFT MACHINE CO.	1
48294	PSI PERIPHERAL SUPPORT	1
4M241	BRAMBLETT ELECTRONICS CO. INC.	1
52202	MODULAR DEVICES INC	1
53753	UNION FED CORR INSTITUTION	2
54736	UNICOR	2
55928	FEDERAL PRISONS INDUSTRIES	1
56132	REYNOLDS AND TAYLOR INC.	1
58163	NOAH HOWDEN INC.	1
61453	R.S. MICROWAVE CO INC.	1
62313	VOUGHT CORP MLRS DIV	2
62445	DEUTZ CORP	1
64059	LTV AEROSPACE & DEF CO	8
66015	HDL RESEARCH LAB INC	1
6X173	WILDWOOD ELECTRONICS INC.	2
70898	BEECH AIRCRAFT CO	5
80378	VOUGHT CORPORATION	2
82577	HUGHES AIRCRAFT CO	14
92059	RAYMOND ENGINEERING INC	2
94580	HONEYWELL INC	1
95542	NORDEN SYSTEMS	1
96214	TEXAS INSTRUMENTS CORP.	2
98284	HUMPHRY INC	1
98659	COMPUTER INSTRUMENTS CORP	2
9N498	LAND L PRECISION MACH CO	1
XXXXX	AMOS METAL PRODUCTS	1

CONTRACTOR - QUALITY DEFICIENCY REPORT SUMMARY
AMSMI-QA-CF

SUMMARY PERIOD : AUGUST 1989
(as of 22 Sep 1989)

CNTR CODE	CONTRACTOR	OVER 180
01178	LOURDES INDUSTRIES, INC.	1
03104	TELEDYNE CAE	1
04939	MARTIN MARIETTA AEROSPACE CORP	7
05716	RAYTHEON CO	9
09087	INFODEX, INC.	1
14339	GENERAL RELIANCE CORP	1
15090	HUGHES AIRCRAFT CO	3
17029	A C INC.	1
19605	INSCOM ELECTRONICS CORP	3
19710	MPC PRODUCTS CORP	1
1EV35	DENHAM MACHINE & TOOL	1
1HE79	RAYTHEON	1
1L007	SPECIAL PROJECTS INC.	1
22255	PRECISION LAPPING & OPTICAL	1
22830	SYMETRICS INDUSTRIES INC.	1
23302	S W ELECTRONICS & MFG CO.	1
24321	CONDOR PACIFIC INDUSTRIES INC	1
26530	PBR ELECTRONICS INC	4
29055	TRAM ELECTRONICS INC	1
32206	SAVE-O-SEAL	1
34449	B-K MFG CO.	1
3L775	UNIVERSAL TECHNOLOGIES	1
3V055	ANCO MACHINE CO.	2
3Y012	SPACECRAFT MACHINE CO	5
4X740	PREFERRED METALS	1
51753	ARISTA DEVICES CORP	3
54736	UNICOR	1
62313	VOUGHT CORP MLRS DIV	1
62445	DEUTZ CORP	3
62502	GOODMAN BALL, INC.	1
62810	APACHE ENTERPRISES INC	1
62983	VICKERS INC.	1
64031	ARRAL INDUSTRIES	1
64143	BESAFA INC	1
64609	BANNER MACHINE CO.	1
6K329	U.S. SMALL BUSINESS ADMIN	4
6X173	WILDWOOD ELECTRONICS INC.	1
70898	BEECH AIRCRAFT CORP.	1
7N710	CONSOLIDATED INDUSTRIES INC.	3
80378	VOUGHT CORPORATION	30
82577	HUGHES AIRCRAFT CO	13
8T985	SWAIM MACHINE CO	1
91196	FISHER ENGINEERING INC	1
92059	RAYMOND ENGINEERING INC	3
92837	UNICOR	7
94580	HONEYWELL INC	1
95542	NORDEN SYSTEMS	1
96214	TEXAS INSTRUMENTS	7

96906	MILITARY STANDARDS PROMULGATED	1
97928	DEUTSCH FASTENER CORP	1
98453	NORTHROP CORP	2
98659	COMPUTER INSTRUMENTS CORP	1
98889	THE TALLEY CORPORATION	1

CONTRACTOR - QUALITY DEFICIENCY REPORT SUMMARY
AMSMI-QA-CF

SUMMARY PERIOD : AUGUST 1989
(as of 22 Sep 1989)

CNTR CODE	CONTRACTOR	OVR 180 CLOSED
00641	FORWAY INDUSTRIES INC	1
04939	MARTIN MARIETTA AEROSPACE CORP	2
05716	RAYTHEON CO	1
0AH14	DEFENSE TECHNOLOGIES INC.	1
13310	AEROJET SOLID PROPULSION CO.	1
15984	MILAN BOX CORP	1
19976	UNICOR	1
1A228	ABSTON MACHINE SHOP INC	2
1BG56	COMMAND TECHNOLOGY - SALT LAKE INC.	1
1CG31	J & J PRECISION MACH SHOP	1
265[illegible]	PBR ELECTRONICS INC	2
34449	B-K MFG CO	2
4K245	SMALL BUSINESS ADMINISTRATION	2
51215	ROCKWELL INTERNATIONAL CORP.	1
52202	MODULAR DEVICES INC	1
63911	MILITARY AIRCRAFT SALES INC	2
64448	R & D MACHINE INC	1
6X173	WILDWOOD ELECTRONICS INC	1
7T482	MILITARY AIRCRAFT SALES CO	1
80378	VOUGHT CORPORATION	1
81039	PLESSEY DYNAMICS CORP	1
82577	HUGHES AIRCRAFT CO	3
99251	LITTON PRECISION PRODUCTS	1
9Y279	RPL ELECTRONICS CO	2

2.4.6 12 Month QDR Summary

Selection of the number "4" from the Empress 32/ASCII Matrix File Menu activates the source code (Appendix G) necessary to produce the 12 Month Quality Deficiency Status Report (Figure 9) for all weapon systems. This option varies from the previous three choices since the program will produce a matrix consisting of the joined grand totals derived for all the weapon systems. This program module is designed to perform a roll-off feature enabling the output matrix to always end with the current month in the far right column. When the program is executed, the prompt will ask the user to enter the input data exactly the same as before (Figure 10) with all source code changes remaining transparent. The resulting output is constructed to add the current month's data and automatically roll-off the data twelve months back. Therefore, each generated matrix will consist of a summary detailing twelve months of QDR milestone information. Due to this module's requirement to have access to input files to accomplish the roll-off effect, caution should be exercised to not accidentally delete or alter any of the previous twelve months' data files (designated by the file extensions .CFO and .t).

```
                QDR Status Report Matrix

        Enter Matrix Filename : 12mo
        Enter Beginning Date (mmddyyyy) : 08011989
        Enter Ending Date (mmddyyyy)    : 08311989
```

Figure 10

2.4.7 File Downloading Instructions

he selection of the code "b" from the main menu will present the user with the QDR Status Barchart Menu (Figure 11).

```
                QDR STATUS BARCHART MENU

                        1-Help
                        2-Main Menu

Enter Choice:
```

Figure 11

AMSMI-QA-CF
WEAPON SYSTEM CODE: ___ALL_SYSTEMS____ QUALITY DEFICIENCY REPORT STATUS
Date: __22_Sep_1989__

ACTIVITIES	SEP 1988	OCT 1988	NOV 1988	DEC 1988	JAN 1989	FEB 1989	MAR 1989	APR 1989	MAY 1989	JUN 1989	JUL 1989	AUG 1989
OPEN BEGIN OF PERIOD	421	439	460	464	415	448	440	477	542	509	535	515
RECEIVED DURING PER	61	72	60	46	70	69	119	131	68	94	72	96
REOPENED DURING PER	0	0	0	0	0	0	0	0	0	0	0	0
CLOSED DURING PER	35	34	30	49	18	63	80	46	87	68	63	96
TRANSFERRED DURING PER	22	15	24	50	18	8	9	8	9	17	14	21
OPEN END OF PERIOD	439	462	468	415	449	447	477	554	526	535	530	509
1 - 119 DAYS	188	200	177	153	168	161	214	308	283	296	262	218
120 - 180 DAYS	63	66	82	70	61	60	49	38	49	57	78	116
OVER 180 DAYS	188	196	209	192	220	226	214	208	194	182	190	175
OVER 180 CLOSED	13	18	14	19	9	27	36	26	40	21	20	36

FIGURE 9

This menu allows the user to select a help screen option or return to the main menu. The menu source code is attached as Appendix H. The help screen option will provide the user two sequential sets of instructions that pertain to the procedure of transporting the QDR barchart ASCII matrix from the host Unisys 5000 to the local Macintosh.

The first instruction set (Figures 12) details a four step process that enables the user to transfer the ASCII file from the Unisys to the Macintosh utilizing the Kermit file transfer protocol.

Sperry/Macintosh ASCII File Transfer Instructions		
1.	At the Unix prompt,	Type : kermit
2.	When the prompt 'C-Kermit>' is present,	Type : send filename enter
3.	To set up Red Ryder,	move pointer to the top status bar, drag on File, and Select : receive file - Kermit Enter : receive filename
4.	When the downloading procedure is completed,	move pointer to the top status bar, drag on File, and Select : quit

Figure 12

The second instruction set describes, in two parts (Figures 13 and 14), a fourteen step process permitting the user to place the transferred file into a renamed copy of the master excel template file for subsequent creation of the weapon system barcharts.

Macintosh ASCII/Excel File Placement Instructions

1. Locate the folder containing the QDR related files.
2. Move pointer to the 'QDR_MASTER_ECL' excel file and click. (Note: NEVER alter or lose the 'QDR_MASTER_XCL' file!)
3. Move pointer to the 'copy of QDR_MASTER_XCL' excel file and DUPLICATE.
4. Move pointer to the 'copy of QDR_MASTER_XCL' excel file and rename it (e.g., '3Q87_3Q89').
5. Move pointer to the new renamed excel file and select (i.e., double-click) to open.
6. Once opened and inside the spreadsheet, move pointer to the fiscal year labels and update if necessary.
7. Move pointer to cell B1, click, and while holding down the mouse button, drag the pointer to cell AB243 (243R x 27C).

Figure 13

Macintosh ASCII/Excel File Placement Instructions (Con't)

8. Move pointer to MENU BAR, click on FILE, choose OPEN and select the ASCII downloaded file (e.g., 3Q87_3Q89.mac).
9. Once opened, move pointer to cell A1, click, and while holding down the mouse button, drag the pointer to cell AA243.
10. Move pointer to the MENU BAR, click on EDIT, and choose COPY.
11. Move pointer to box at top left-hand corner and click to close the file. Respond to the displayed message by choosing 'Save Formatted Values'.
12. Move pointer to Menu Bar, click on EDIT, and choose PASTE.
13. When successfully loaded, move pointer to box at top left-hand corner, click to close the file, and choose the SAVE option.
14. This completes the required steps for preparing the weapon system data to create the Macintosh generated barcharts.

Figure 14

Once the QDR weapon system data are placed into the renamed excel file, the weapon system barcharts are ready to be created. Some weapon system barchart files exist and can be duplicated for adding or editing purposes. The crucial step that must be achieved to print a chart (Figure 15) is to link the weapon system barchart file with the current excel weapon system data file in use.

The following steps describe how to link the barchart file with the renamed excel template file:

1. Move the mouse pointer to the folder which contains the barchart file to be selected and double-click.

2. Choose the option "OK" in response to the message "Update References to Non-Resident Sheets?"

3. Choose which weapon system ASCII file is desired for linking by selecting COPY.

The barchart file is now linked to the weapon system data which was downloaded and placed in the renamed excel template file.

2.4.8 Special Functions Menu

The selection of the code "c" from the main menu will present the user with the Special Functions Menu (Figure 16). This menu allows the user to select a custom date reorder feature or return to the main menu. The source code for this menu is attached as Appendix I.

```
                    SPECIAL FUNCTIONS MENU
_____________________________________________________________

                          1 -  Date Reorder
                          2 -  Main Menu

Enter Choice :
```

Figure 16

2.4.9 Reorder Menu

Selection of the number "1" from the Special Functions Menu will present the user with the Reorder Menu (Figure 17).

CHAPARRAL QDR STATUS

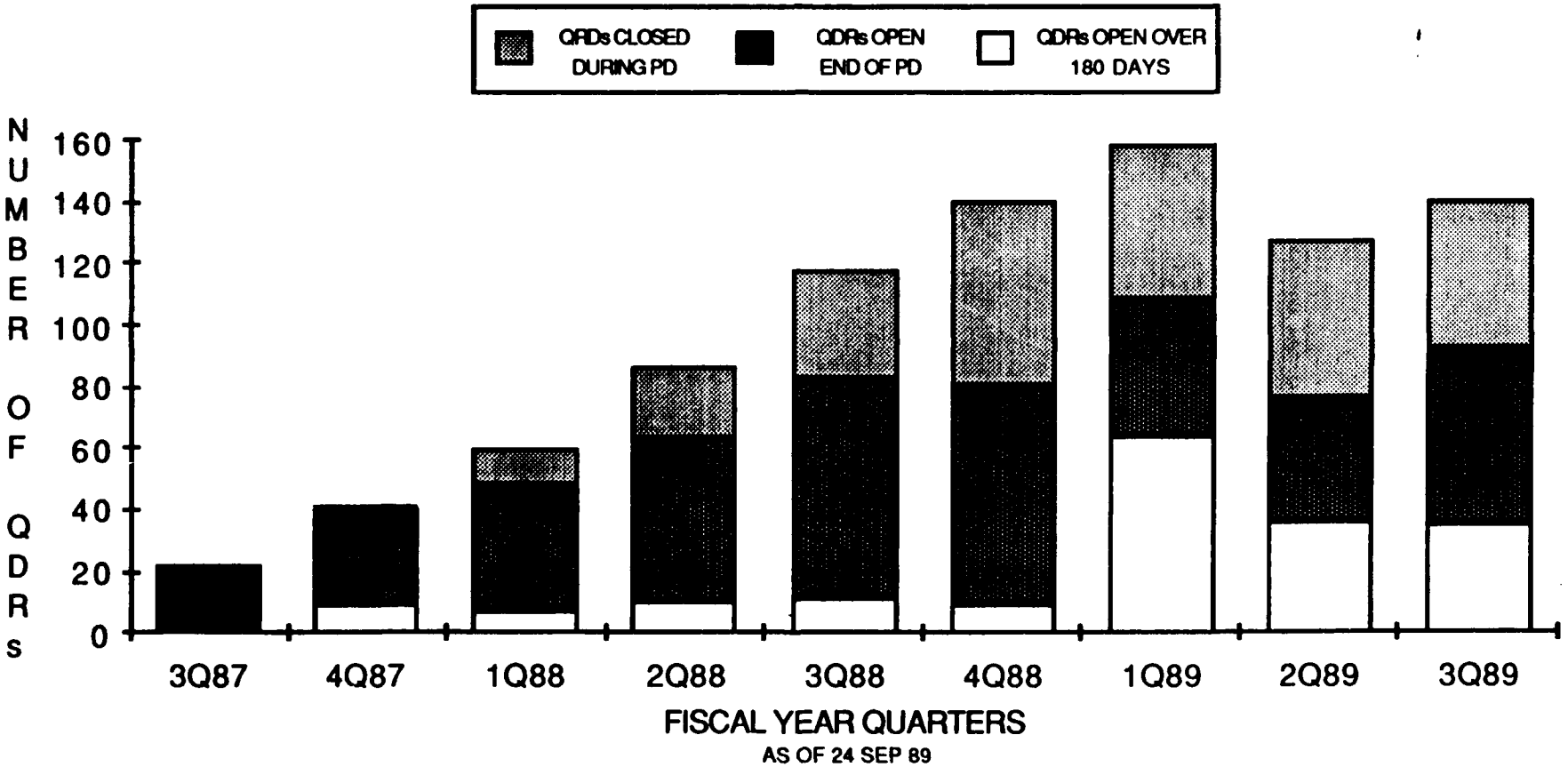


FIGURE 15

```
                    REORDER MENU

                    1 -  Input Data
                    2 -  File Status
                    3 -  Specail Functions Menu

Enter Choice:
```

Figure 17

Selecting number "1" at this point will activate the source code (Appendix J) necessary to execute the date reorder function (Figure 18).

This utility allows the user to split, reorder, and merge a specified field in a database table.

```
          Shell Script Parameters :

          p1-  table name with fields separated by ^V.
          p2-  field position in table.
          p3-  first cut starting position in field.
          p4-  first cut character length in field.
          p5-  second cut starting position in field.
          p6-  second cut character length n field.
          p7-  table name of reordered field.

          Enter p1              : cd8nward_demo
cd8nward_demo
          Enter p2              : 4
4
          Enter p3 and p4       : 1 4
1 4
          Enter p5 and p6       : 5 4
5 4
          Enter p7              : xcd8nward_demo
xcd8nward_demo
```

Figure 18

The date reorder function prompts the user to make the following inputs: 1) the name of the file containing the data to be reordered, 2) the field position to perform the reorder, 3) the first beginning character position within the field and the number of characters to be extracted together, 4) the second beginning character position within the field and the number of characters to be extracted together, 5) the file name that will contain the new reordered data.

Selection of the number "2" from the Reorder Menu will display a status screen of the file that was manipulated and re-created.

Selection of the number "3" will return the user to the Special Functions Menu.

This code was originally developed to correct the date mismatch between S2K and Empress. In addition to date reorder, the reorder feature can be used to perform customized operations for future applications.

2.5 Prepare Sort Routines

The process needed to sort the QDR data by the below characteristics has been developed and proven using the procedure discussed in paragraph 2.4.

The main problems for implementation on the present system are the same as mentioned earlier: data located in too many tables and date field not in the proper format for Empress query.

2.5.1 Weapon System

This sort is used to develop both the QDR summary and status reports as requested.

2.5.2 Action Officer

This sort produces a report that presents the QDR workload in the same format as the Weapon System summary, only by action officer.

2.5.3 Type Deficiency Report

Not implemented.

2.5.4 Age of QDR

Not implemented.

2.5.5 Origin of QDR

Not implemented.

2.5.6 Contractor

The report processing is the same as for Weapon System and Action Officer and depicts workload by contractor format was modified due to the number of unique contractors.

2.6 Prepare Outgoing Letters and Reports

The sponsor requested that an electronic mail system be developed that would allow CFO and QM to send required correspondence to each other for comment and concurrence. It was further requested that an alerting or flagging system be established that would alert

action officers and managers that one of the letters was due to be dispatched within the next 60, 30, etc., days.

The following paragraphs discuss three utilities developed under this task.

2.6.1 Memo Utility for DRS Update

The purpose of the memo utility (Appendix K) is to allow each individual action officer to send CFO a **REMARK** or a **CLOSEOUT** action memo through electronic mail on the Unisys 5000 computer. The user will input data with the aid of entry prompts. After all data are entered, the screen will refresh with the input data, and the user will have the option of making any changes. If no changes are needed, a copy of the screen will be automatically sent to CFO by the Unix mail facility. A copy of the **new** mail will be dumped out of the mailbox into a Unix file and printed. These will be the sheets for which data is entered into key-plus. Mail can be read many times during the day, but each time a new output file must be created.

To access the MEMO program type:

```
$memo <return>
```

Screen one appears:

```
DRS UPDATE MEMO

1. REMARK

2. CLOSEOUT

3. QUIT

ENTER UPDATE ACTION: __
```

If **1** is entered, the following entry screen appears:

```
                    DRS UPDATE ACTION

DOCUMENT MANAGEMENT NUMBER: _________   ACTION OFFICER: ____

ACTION PROCESS CODE: __          DATE OF ACTION (MMDDYY): ______

DODAAC SPT POINT: _____

ACTION: ____________________________________________________
```

After entering the data at each prompt, a carriage return will move the cursor to the next field. The user will not be able to back up after a carriage return has been entered. Corrections can be made after all entries have been made the first time. Only one line is allowed for the entry **ACTION** due to the fact that only 50 characters are allowed for this field in key-plus.

After the data for **ACTION** has been entered, the following screen appears with an example case:

```
                    DRS UPDATE ACTION

1) DOCUMENT MANAGEMENT NUMBER: P19L00023 2) ACTION OFFICER: GSR

3) ACTION PROCESS CODE: B                4) ACTION DATE: 091689

5) DODAAC SPT POINT: EXAM

6) ACTION: The letter was sent to the investigation officer._____


                ARE ENTRIES CORRECT (Y/N) ?: __
```

If a "Y" or "y" is entered, all the data on the screen are echoed to a file with the name **<dmn.apc>** (document management number.action process code). In the example case, the file would be called (P19L00023.B). The file P19L00023.B is now mailed to CFO with the message

```
**** SENDING MAIL ****
```

appearing on the screen. Once the mail is sent the program returns to the beginning screen, which asks the user to enter a REMARK or CLOSEOUT action.

If an "N" or "n" is entered, the following prompt appears:

```
ENTER NUMBER TO CHANGE: __
```

The user must enter a number in the range [1-6], or nothing will happen. When a number in the range is entered, the cursor will move to the corresponding field. The user must now reenter the entire field, or some of the characters will be lost. After the changes have been made, a carriage return will bring up the prompt

```
ARE ENTRIES CORRECT (Y/N) ?: __
```

This process will continue until a "Y," "y" or an abort key is pressed. When a "Y" or "y" is entered; a

```
**** SENDING MAIL ****
```

message appears and the beginning program screen appears.

If a **2** is entered at the DRS UPDATE MEMO MENU, the following entry screen appears:

DRS CLOSEOUT ACTION

DOCUMENT MANAGEMENT NUMBER: _____ ACTION OFFICER: ____

TYPE QDR CODE: ____ FINAL DISPOSITION CODE: ____

ACTION PROCESS CODE: __ DATE OF ACTION (MMDDYY): _____

CARD 01: _____
CARD 02: _____
CARD 03: _____
CARD 04: _____
CARD 05: _____
CARD 06: _____
CARD 07: _____
CARD 08: _____

If there are no entries for a particular field, press the <return> key and the cursor will move to the next field. When all fields have been entered, the screen will refresh with the prompt:

```
ARE ENTRIES CORRECT (Y/N) ?: __
```

The process is now the same as the REMARK action routine.

If a **3** is entered in the opening menu, the screen is cleared and the program exits back to a Unix system prompt.

DRS MEMO PROGRAM (CFIM PORTION)

Login to Sperry 5000:

```
pad2
UNISYS 5000 Series
Login: cfim <CR>
Password:       <CR>


Enter terminal type: ____ <CR>
```

If there is no new mail, the Unix system prompt will appear. If there is new mail, the message

```
YOU HAVE MAIL
```

will appear and control is transferred directly into the mail facility.

The **mailx** screen now appears with all mail.

```
mailx version 3.0  Type ? for help.
"usr2/acct/cfim": 4 messages 4 new
>N  1     gross     Tues Sep 21    08:09          8/155
 N  2     gross     Tues Sep 21    08:11          10/185
 N  3     wblanton  Tues Sep 21    09:11          8/148
 N  4     sscott    Tues Sep 21    10:13          15/275
?
```

The user may read the mail by typing any or all of these messages to the screen.

`? p <CR>` -- this types the current message to the screen

`? p * <CR>` -- this types all messages to the screen

`? p 3 <CR>` -- this types message number 3 to the screen

`? p :n <CR>` -- this types all new messages to the screen

`? p gross <CR>` -- this types all messages from user gross

The user should now extract all new mail into an operating system file with the current date (MMDDYY) as the filename. If mail is extracted more than once during the same day, the second file should have an a,b,c, etc. appended on the end of the filename (MMDDYYa, (MMDDYYb, etc.). If an a,b,c, etc., is not appended at the end of the filename, the original file will be overwritten.

? s :n <filename> <CR> --extracts new mail and places it in <filename>

The user should now exit the mail facility.

```
? q <CR>    -- this quits the mail facility.
            -- all mail that has not been read is now flagged with a U (unread).
            -- if mail has been read, the messages are taken out of the mail facility
               and placed in the operating system file mbox.
```

All mail that has been read and extracted from the system should be deleted once a day, preferably after the extraction.

```
? d 1-7 <CR>      -- this deletes messages 1 thru 7
```

This deletion should be done to ensure that messages do not get extracted more than once. When the user quits the mail facility, the messages are removed from the mailbox.

To enter mail facility after exiting, type "mailx" at the Unix system prompt.

2.6.2 QDR Alerting System

The sponsor requested that an alerting system be developed for alerting action officers and others when interim or final replies to the QDR originator are due. The sponsor also wanted the system to alert the action officers and others when other specific actions were due. These actions include requests for support, follow-ups, etc. Automatic notification by the computer was requested.

The requirement has been accomplished for computer alerting of personnel when replies to the QDR originator are due. There was insufficient time left on this task order to accomplish the other alert actions. However, these additional alert requirements could be accomplished with procedures and software very similar to that developed for replies to the QDR originator, which is discussed below and is extensively documented in Appendix L of this report.

2.6.2.1 Objectives of the Alerting System

The specific objectives of this system were developed in discussions with the chiefs of the Customer Feedback Office and the Quality Requirements and Data Evaluation Branch in PAD. The system was to alert the various action officers two weeks in advance whenever an open QDR assigned to them required an interim or final reply to the QDR originator. These are required within 90 days after receipt of the QDR and every 60 days (or less) thereafter until the QDR has been resolved. A follow-up notice was to be sent to the action officer two weeks after the due date, if necessary. A weekly listing (by action officer) of QDRs overdue for reply to the originator was to be sent to the action officer supervisor. A monthly listing was to be sent to the Customer Feedback Office of all QDRs still overdue for reply to the originator at the end of the month. These alert notices and listings were to be sent via electronic mail to the specific personnel concerned.

2.6.2.2 Operation of the System

The QDR alerting system is installed on the PAD2 Unisys 5000/80 computer located in the Customer Feedback Office. It is used to track processing activities associated with the Deficiency Reporting System (DRS) which has been loaded into Empress database tables on

the PAD2 Unisys computer. As illustrated in Figure 19, the computer, terminals, and interconnecting cables provide a network for disseminating QDR information directly from the computer to each person involved in processing QDR data. Access to any of these data is restricted by use of user IDs and login passwords, as well as the provision for restriction of "read" privileges on user owned files. The Empress database management system also provides additional restrictions on access to database tables.

Figure 20 shows the structure of the DRS database as it is currently installed on the Unisys computer. A detailed description of each table is shown on pages 1 through 6 of Appendix L to this report. The data contained in tables *s2krg0* through *s2krg1100* are essentially as defined in AMC Pamphlet 702-32. These tables will automatically be updated once a week (at night) by the computer, in accordance with the schedule contained in the *cron* table (see page 7 of Appendix L). Seven additional tables have been added to the database to aid in tracking QDR processing actions. Data will be entered automatically into these tables in accordance with instructions in the *cron* table. The exceptions to this are the *moraction* and *action_officers* tables, which are discussed further below.

The primary data elements being searched by the alerting system are the document management number, the report status code (open, closed, etc.), the item name, the date received and the action officer code. These elements are in the *s2krg0* table. Action process codes and associated dates are contained in the table *s2krg200*. Since the System 2000 (S2K) database limited this table to a maximum of 9 entries (records) per QDR, the existing S2K tables cannot contain all of the actions and codes (there are currently about thirty different codes) which could be applicable to a given QDR. This means that the standard S2K tables can not be used to provide a full history or track of actions taken in processing a given QDR. Therefore, the table called *moraction* was added to the Empress database to handle the overflow of action process data from the S2K system. There is no practical limit to the number of action process code records per QDR which may be added to this table. At the present time, these additional action process code records must be entered interactively by the CFO data entry personnel.

The *action_officers* table was added to facilitate management of the alerting system by the DRS database manager. This table contains the currently valid action officer codes and the associated action officer names and login IDs for processing the alert notices via electronic mail. This table is also updated interactively by the CFO data entry personnel. This table is used in the Empress command files (see pages 14 and 16 of Appendix L) to determine which action officer codes are searched in updating the *reply_due* and *reply_overdue* tables. Thus, non-PAD codes may be skipped. This table was also intended to provide the database manager with external control over the alerting system by providing for addition and deletion of action officer codes, login IDs, etc. However, time did not permit incorporating the use of this table into the shell script programs (see pages 18 and 20 of Appendix L), which identify the action officers and direct the mailing of the alert notices. This should be accomplished in a future task order to simplify management of the alerting system.

PAD COMPUTER NETWORK FOR QUALITY DEFICIENCY REPORT (QDR) ALERTING SYSTEM

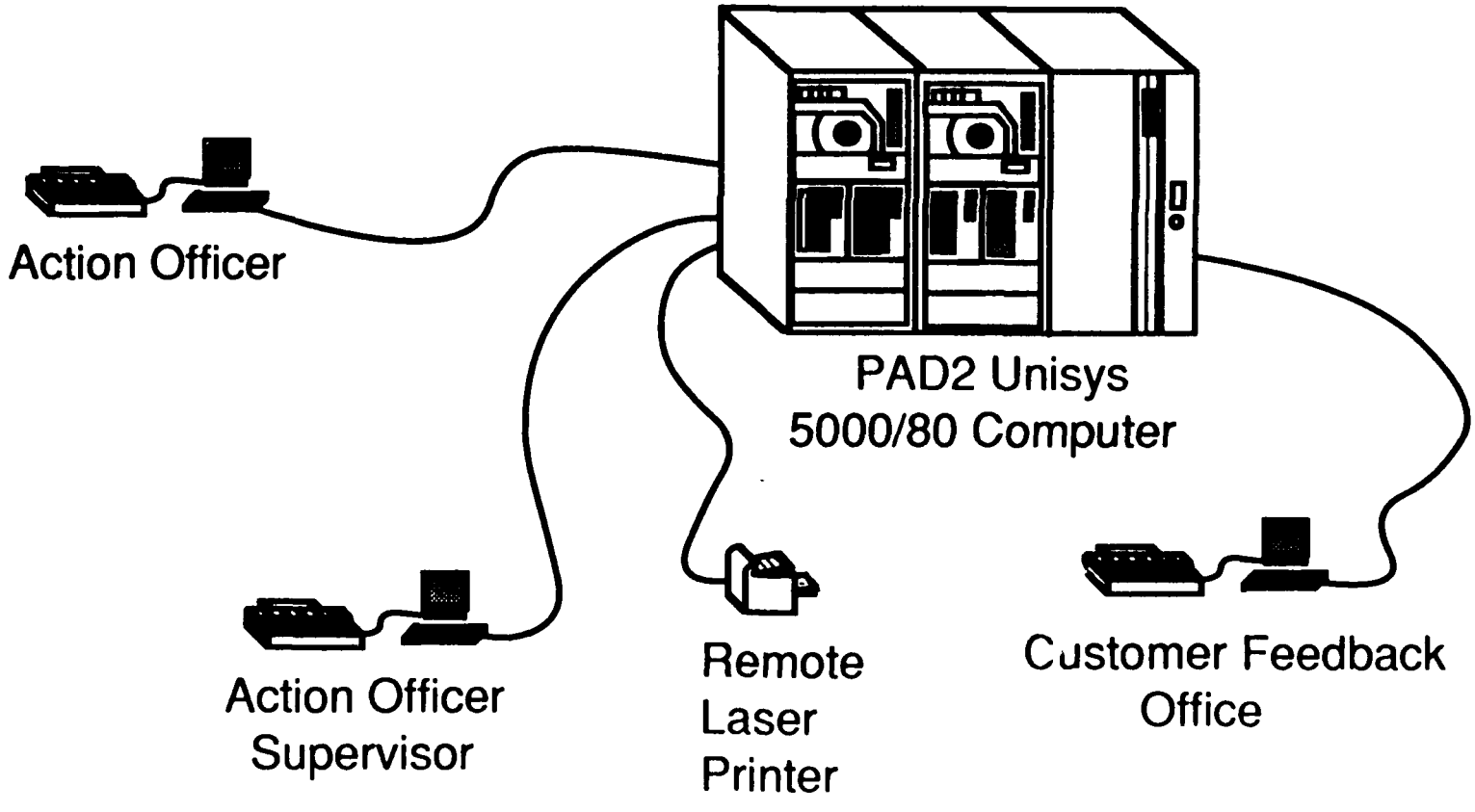


FIGURE 19

STRUCTURE OF DEFICIENCY REPORTING SYSTEM (DRS) DATABASE

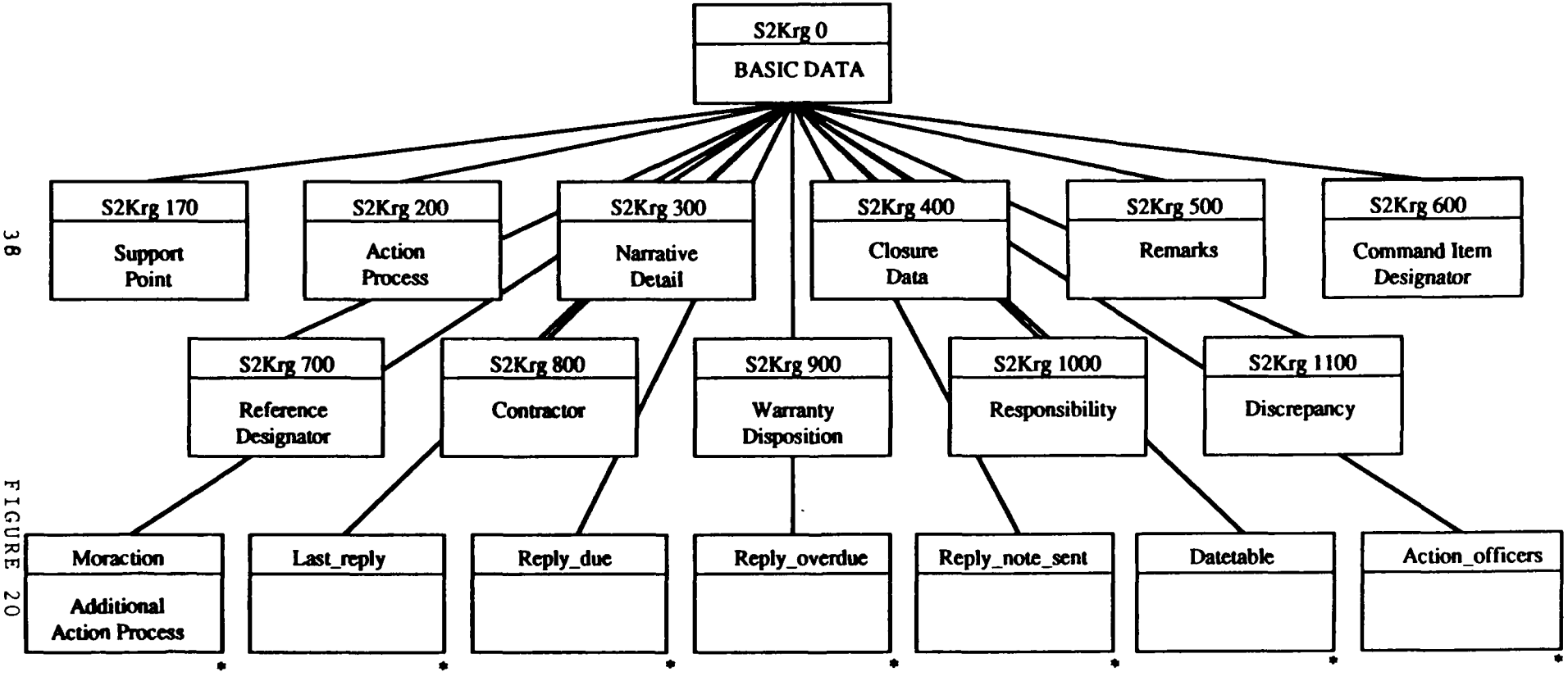


FIGURE 20

* identifies new tables added for QDR alert system

The *last_reply* table contains document management numbers of open QDRs and the last date on which an interim or final reply was sent to the QDR originator. This data provides more depth to the alert messages sent by the system. The table is updated automatically by the computer every morning. The table called *datetable* contains the current date (used for calculating age of the QDRs, etc.) and is updated daily by the computer. The *reply_due* table is automatically updated every night and contains data to identify QDRs which are now overdue (for reply to the QDR originator) or will become overdue during the next two weeks. However, an alert notice for any particular QDR will be sent only once (this notice will be recorded in the *reply_note_sent* table). The action officer will not be notified again concerning that particular QDR until after it has become at least two weeks overdue. The *reply_overdue* table contains data to alert the action officers as to which of their QDRs are already at least two weeks overdue for reply to the QDR originator. This table will be updated automatically every Monday morning and alert notices sent to each action officer. An overdue QDR will keep appearing in the "reply overdue" listings each week until the QDR is resolved.

As indicated above, operation of the alerting system is essentially automatic, with actions taken in accordance with the schedule in the *cron* table. Figure 21 shows the various actions taken by the computer in operation of the alerting system. The Unisys operating system continuously examines the various *cron* tables and executes any jobs in these tables according to the schedule in the table for each job. If the computer should be down (power failure, maintenance, etc.) when a particular job is scheduled in the *cron* table, the *cron daemon* will attempt (when the computer is running normally again) to catch up and run all the jobs it missed while the computer was down. In this case, the computer is directed to update the DRS tables (with new data downloaded from the S2K database) every Thursday night at 8 p.m. (see the *cron* table listing on page 7 of Appendix L). The *reply_due* table is updated every morning at 2 a.m. The *reply_overdue* table is updated every Monday morning at 2:30 a.m. Daily notices (i.e., reply due within the next two weeks) to the action officers are sent every morning (including weekends and holidays, if the computer is not turned off) at 4 a.m. Weekly notices (for replies already overdue) to the action officers are sent every Monday morning at 4:30 a.m. The action officer supervisor is also sent a composite listing (by action officer) of QDRs overdue, every Monday morning at 5 a.m. The Customer Feedback Office is sent a monthly listing of overdue reply items on the first of each month at 5:30 a.m. These alert notices or listings are sent via electronic mail. When the addressees log on to the computer, the mail system will advise them if they have mail. The addressee can view the mail through the OFIS menu and may obtain a hard copy of the mail if desired. Examples of the various alert notices and listings are shown on pages 33 through 42 of Appendix L.

The files listed on pages 8 through 35 of Appendix L constitute the various Unix shell scripts (pages 8 through 13 and 18 through 23), Empress command files (pages 14 through 17) and Empress mwriter (report writer) files (pages 24 through 35) required to operate the QDR alert system. These files all belong to and are utilized by the DRS database administrator. The shell script *update_drs* utilizes Empress commands to download DRS data from the S2K database into temporary "dump" files (tab0, tab170, etc.). Data currently in the DRS *s2krg0*, *s2krg170*, etc., tables (but not in the additional non-S2K tables) is then erased and replaced with data from the "dump" files. The necessary DRS database indexes are then recreated so as to speed up search and retrieval of data. The shell scripts *updt_reply_due* and*updt_reply_od* use Empress commands (i.e., */usr/bin/mscmd*) to update the corresponding database tables via the Empress command files *updt_day_reply* and*updt_wk_reply* .

FLOWCHART FOR DRS ALERT SYSTEM

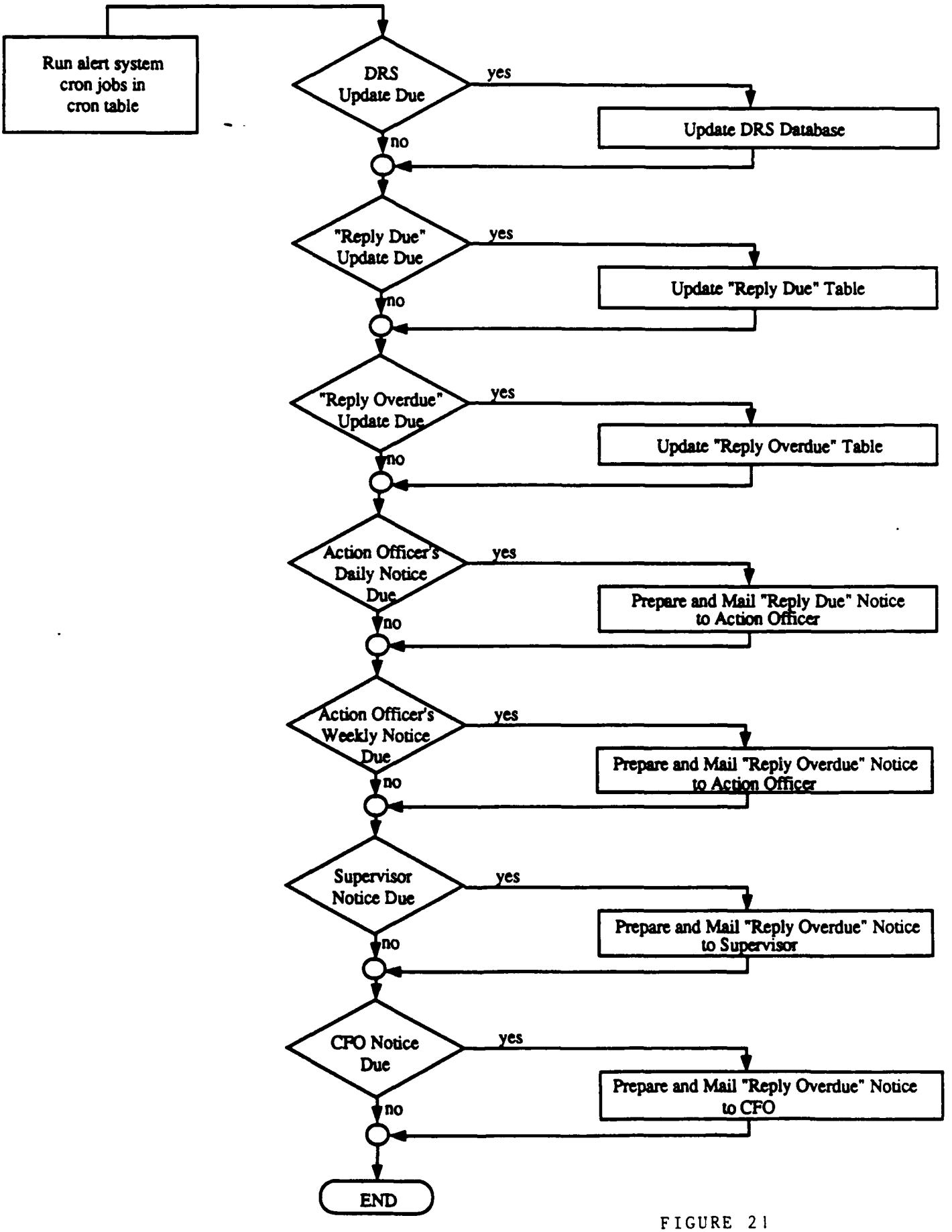


FIGURE 21

The shell scripts *day_reply_note* and *wk_reply_note* are used to send daily and weekly reply notices to action officers. Notices are not sent to an action officer if there are no "reply due" or "reply overdue" QDRs for him at the time these two shell scripts are executed. This is accomplished by use of the "select count" statements to determine if notices are to be sent to the various action officers. Note that the corresponding "reply_due" and "reply_overdue" tables have already been updated at the time these two shell scripts are run. Also note that the action officer codes, names and user IDs are currently built into these shell scripts. Consequently, these shell scripts must be modified if the action officer codes and names are changed (at least until use of the *action_officers* table can be incorporated into these shell scripts). These shell scripts perform their work by executing the Empress mwriter program in conjunction with the mwriter files *reply_due_rpt* and *reply_od_rpt*. The *supv_reply_not* and *cfo_reply_note* shell scripts work in a similar fashion. Any of these shell scripts can of course be run directly by the user when he is at the Unix command level (i.e., has a $ prompt) and has the appropriate Unix and Empress permissions to run these programs.

Examples of the various alert notices and listings are shown on pages 36 through 42 of Appendix L. Each of these notices contain data for one or more open QDRs which are currently due or overdue for reply to the QDR originator. Current data is retrieved from the database and included in these notices, as described above. Values are included for the document management number, the item name, the date the QDR was originally received, the current age of the document and the date the last reply to the originator was sent. The total number of documents included in each alert notice is also shown. The monthly listing mailed to CFO does not identify the action officer but does list each due or overdue QDR and shows the total number of documents.

The format and content of the alert notices are determined by the mwriter files (see pages 24 through 35, Appendix L). These files could be modified slightly to produce different formats and content, if desired. For example, the CFO listings could be modified to include action officer codes. However, extensive changes to the alert notices might require altering the database tables and creation of new Unix shell scripts and Empress command files. This report contains recommendations for further augmentation of the QDR alerting system.

2.6.2.3 User Interface with the System

Since the operation of the QDR alerting system is essentially automatic under the control of the *cron daemon*, the only requirement for normal user interface with the system is to view and possibly print the electronic mail alert messages. At the Unix command level (i.e., at the $ prompt) the user can simply enter the command "mail." However, it is possible that the mail may scroll off the screen before it can be viewed. Using the screen-hold key will help, but the message cannot be scrolled down once it is off the screen. A better approach is to use the "Mail Services" feature in the OFIS menu provided with the Unisys software. The "Get Mail" option should be selected from the submenu. Each piece of electronic mail received but not yet deleted may then be read (press <EXEC>). After the mail has been read and/or printed, it must be deleted (press <DELETE>) if the user does not wish to keep it. A considerable amount of old mail can accumulate in a short period of time if it is not deleted regularly.

The alert messages may be printed (select "SOFT PAGE" while in the "Mail Services" menu) or copied to another file (select "COPY"). If the copy option is selected, the computer will prompt for the directory (identified on the screen as "folder"), and the file name (identified as "document") to which each piece of mail is to be copied. The "Mail Services" software will insert a line ("from cfdba ....") at the top of the copied file. The user may use the *vi* editor or the word processor to delete this extra line if desired. The alert message already has "from" and "to" lines, so this extra line is redundant and should be removed if a neater printout is

desired. After the alert message has been copied, it may be printed on the slaved printer attached to the user's terminal or on the remote laser printer. An improved laser print routine is available for this. The command to be entered at the $ prompt is "laserprint." The laser print routine will then prompt for the name of the file to be printed and for laser printer and font selections. Care must be used, of course, in directing printouts to a remote printer as the outputs may become lost in the printer queue, which could compromise the confidential nature of the alert messages. Generally speaking, most of the daily alert messages should be short and will not need to be printed.

In addition to the role of "user," the DRS database administrator may have an additional interface with the QDR alerting system. As mentioned previously, he may need to use the *vi* editor to change one of the shell scripts or Empress mwriter or command files shown in pages 7 through 35 of Appendix L. All of these files may be found in the *drsdb* directory, with a path name of */usr4/acct/cfdba/drsdb*. The user ID for the DRS database administrator (the "owner" of all the DRS database tables) is *cfdba*. At this point the database administrator will not need to grant database permissions (for the QDR alert portion of the system) to anyone except the CFO data entry personnel, who may be required to insert or update data in the *moraction* and *action_officers* tables.

To insert or update data in the DRS database tables interactively, the following commands are entered at the $ prompt:

cd /usr4/acct/cfdba/drsdb
ms drs

The Empress database system will then respond with a dot (.) prompt. Data may then be inserted into the tables by entering "insert into moraction;" or "insert into action_officers;." The system will then prompt for each value to be entered in the new record to be inserted into the database. The Empress manuals explain in detail how the interactive insert process works. Existing DRS records may be updated interactively in a similar fashion. These two tables (*moraction* and *action_officers*) are the only ones needing data inserts or updates at this time. All other data entry activity for DRS data should be processed with the usual S2K database procedures for the time being.

Of course, any and all data in the Empress DRS database can be selected and viewed or printed by any authorized user (via the *ms drs* command and appropriate Empress select commands). This report contains further recommendations for development of special query screens and other special access reports (for example, where each action officer can see any data on QDRs assigned to him). The QDR alerting system restricts data access by mailing notices only to concerned and authorized personnel.

2.6.3 Laser Printer Utility

The purpose of the laser printer utility (Appendix M) is to route print jobs to individual laser printers connected to the Unisys 5000 located in the Customer Feedback Office. The program lets the user choose which laser printer to send the job and also gives the user the option of choosing one of eight standard fonts (4 Portrait and 4 Landscape). The user will be prompted to give a filename for printing, including path if needed. If the file cannot be found, the program exits out to a Unix system prompt.

To access the print utility type:

```
$prt <return>
```

Screen one appears:

```
LASER PRINTER / FONT PROGRAM

Enter filename : _____________
```

If the file is not found, the following error message appears:

```
**** No file <filename> ****
```

If the file is found, the following printer screen appears:

```
SPERRY 5000 -- LASER PRINTERS

    1) CFLASER
    2) CCLASER
    3) QDLASER

Enter number of laser to print : __
```

A number in the range [1-3] must be entered, or the program will remain at the current prompt. If a valid number is entered, then the following font menu appears:

LASER PRINTER
FONT MENU

	FONT	ORINT	POINT	PITCH
1)	COURIER	P	12	10
2)	PRESTIGE ELITE	P	10	12
3)	PRESTIGE ELITE	P	7.2	16.6
4)	LINE PRINTER	P	9	16.6
5)	COURIER	L	12	10
6)	PRESTIGE ELITE	L	10	12
7)	PRESTIGE ELITE	L	7.2	16.6
8)	LINE PRINTER	L	9	16.6

```
Enter Font Choice : __
```

If a valid font is chosen, the message

```
**** PRINTING IN PROGRESS ****
```

appears on the screen; otherwise, the prompt for a Font Choice will reappear until a valid font is chosen.

The first process in printing the file is to send the escape sequence for the chosen font to the printer. The second process sends the actual file to the printer. After the first file is printed, the user is prompted:

```
PRINT ANOTHER FILE (Y/N)
```

If a "Y" or "y" is entered, the program restarts; otherwise, the program exits to the Unix system prompt.

3.0 Status of Accomplishments

3.1 Menu Driven

The processes discussed in this report are fully operational and ready for installation by CFO.

3.2 Weapon System QDR Summary

This procedure and the next three (Action Officer Summary, Contractor Summary, and QDR Status Summary) are fully operational from a processing stand point. There is one problem that was discovered when the charts were run for August and compared to the July charts (Figures 22 and 23): the OBOP for August does not match the OEOP for July, which is found in the CFO manually prepared charts. The computer only recognized one state for a QDR at a time; it is either open, closed or transferred. All QDRs that are closed or transferred on the first day of the month are counted as closed for that day and not as open at the beginning of the month.

A run of those closed and/or transferred on 1 August 1989 proved that this was the case, Figure 24. Five QDRs were closed and ten were transferred, accounting for the difference of 15 between July OEOP and August OBOP.

```
* run;
select rept_sta_cd from s2krg0 where docu_mgt_no match "P*"
and (rept_sta_cd="T" or rept_sta_cd="C" or rept_sta_cd="R")
and rept_sta_cd_dt >= '19890801' and rept_sta_cd_dt <=
'19890801' and act_pt != "QACF";

T
C
C
C
C
C
T
T
T
T
T
T
T
T
T
*
```

Figure 24

QUALITY DEFICIENCY REPORT SUMMARY
AMSMI-QA-CF

SUMMARY PERIOD : JULY 1989
(as of 14 Sep 1989)

WEAPON SYSTEM	OBOP	RCVD	CLOSED	TRANSF	OEOP	1-119 DAYS	120-180 DAYS	OVER 180	OVR 180 CLOSED
OH-58D	9	3	0	2	10	10	0	0	0
CHAPARRAL	105	7	8	8	96	34	16	46	4
HAWK	64	9	8	0	65	30	15	20	3
TARGETS	19	2	1	0	20	16	1	3	1
REDEYE	5	0	0	0	5	1	0	4	0
STINGER	10	6	0	0	16	13	1	2	0
PATRIOT	51	4	21	0	34	19	8	7	4
FAAR	12	0	1	0	11	7	3	1	1
MLRS	66	6	5	1	66	13	11	42	4
LANCE	5	2	2	0	5	2	0	3	0
HELLFIRE	7	4	0	0	11	5	4	2	0
LCSS	2	0	2	0	0	0	0	0	0
TOW	63	3	2	0	64	22	11	31	0
PERSHING II	19	0	0	0	19	6	2	11	0
DRAGON	22	1	3	2	18	16	1	1	0
TOW BRADLEY	28	0	2	0	26	21	1	4	2
GLLD	1	0	1	0	0	0	0	0	0
MULE	0	0	0	0	0	0	0	0	0
THERM IMAGE	17	5	7	0	15	12	1	2	1
TADS	5	8	0	1	12	12	0	0	0
CALIBRATION	3	0	0	0	3	3	0	0	0
APACHE	1	9	0	0	10	9	0	1	0
AN/TSQ-73	1	0	0	0	1	1	0	0	0
TOW COBRA	20	1	0	0	21	8	3	10	0
FIST-V	0	1	0	0	1	1	0	0	0
UNKNOWN	0	1	0	0	1	1	0	0	0
TOTAL	535	72	63	14	530	262	78	190	20

FIGURE 22

QUALITY DEFICIENCY REPORT SUMMARY
AMSMI-QA-CF

SUMMARY PERIOD : AUGUST 1989
(as of 14 Sep 1989)

WEAPON SYSTEM	OBOP	RCVD	CLOSED	TRANSF	OEOP	1-119 DAYS	120-180 DAYS	OVER 180	OVR 180 CLOSED
OH-58D	10	3	0	1	12	4	8	0	0
CHAPARRAL	91	3	6	3	90	16	24	50	1
HAWK	65	5	17	0	53	20	11	22	3
TARGETS	20	7	2	0	25	14	8	3	0
REDEYE	5	0	1	0	4	0	1	3	1
STINGER	16	2	1	3	14	8	4	2	0
PATRIOT	34	14	19	1	28	20	4	4	3
FAAR	11	0	2	0	9	3	5	1	0
MLRS	65	4	9	0	61	11	13	37	7
LANCE	5	3	3	2	3	3	0	0	3
HELLFIRE	11	3	6	0	8	5	3	0	3
LCSS	0	1	0	0	1	1	0	0	0
TOW	64	9	16	0	57	18	12	27	10
PERSHING II	19	0	0	0	19	3	5	11	0
DRAGON	18	3	3	1	17	17	0	0	1
TOW BRADLEY	26	11	1	0	36	22	9	5	0
GLLD	0	0	0	0	0	0	0	0	0
MULE	0	0	0	0	0	0	0	0	0
THERM IMAGE	15	2	2	0	15	13	1	1	2
TADS	12	11	0	0	23	21	2	0	0
CALIBRATION	3	0	1	0	2	0	2	0	0
APACHE	2	0	0	8	2	1	0	1	0
AN/TSQ-73	1	0	1	0	0	0	0	0	0
TOW COBRA	21	10	6	0	25	13	4	8	2
FIST-V	1	2	0	0	3	3	0	0	0
UNKNOWN	0	2	0	1	2	2	0	0	0
TOTAL	515	95	96	20	509	218	116	175	36

FIGURE 23

3.3 Action Officer QDR Summary

See 3.2.

3.4 Contractor QDR Summary

See 3.2. In addition to the closed or transferred problem discussed in 3.2, Contractor QDR Summary has the additional problems discussed below. (Refer to Figure 25 for examples of each problem cited.)

Typographical errors in entering contractor name (items 17, 18, 21), variations in the same name (items 17, 18, 20 - 25, 37, 38, 42 - 44, 54 - 57), multiple contractors listed for same contractor code (items 10 - 13, 41 - 44) and several QDRs have no contractor code identified.

3.5 QDR Status Summary

See 3.2.

3.6 Bar Charts

The procedures necessary to produce these charts on the CFO Macintosh SE are fully operational and ready to install.

3.7 Electronic Mail

The processes discussed in this report are operational and being tested by PAD, CFO and QM personnel.

3.8 Alerting System

This procedure is operational and ready to be installed.

3.9 Graphics Capability

The contractor has developed programs that enable CFO personnel to utilize the capabilities of both the Macintosh SE with laser printer and the other laser printers slaved and remote to the Unisys 5000.

1	00641	FORWAY INDUSTRIES INC
2	01178	LOURDES INDUSTRIES, INC.
3	01351	SOUTHERN GEAR & MACHINE
4	01848	TARTAN INDUSTRIES
5	02101	HTL ELECTRO KINETICS
6	02184	OBRIEN GEAR COMPANY
7	02266	MORTON THIOKOL INC.
8	02266	THIOKOL CHEMICAL CORP
9	02418	KEYSTONE ENGINEERING CO.
10	02731	HUGHES HELICOPTERS
11	02731	MCDONNELL DOUGLAS HELICOPTER CO
12	02731	MCDONNELL DOUGLAS HELICOPTER CO.
13	02731	MCDONNELL-DOUGLAS HELICOPTER CO
14	02987	BENDIX CORP
15	03104	TELEDYNE CAE
16	03550	VANGUARD ELECTRONICS CO INC
17	03860	BOWMAN/ALI INC
18	03860	BOWMAR/ALI INC
19	04597	PROJECTS UNLIMITED
20	04939	
21	04939	MARIETTA MARIETTA AEROSPACE CORP
22	04939	MARTIN MARIETTA AEROSPACE CORP
23	04939	MARTIN MARIETTA CORP
24	04939	MARTIN MARTIETTA AEROSPACE CORP.
25	04949	MARTIN MARIETTA AEROSPACE CORP
26	04971	EMERSON ELECTRIC COMPANY
27	05236	JONATHAN MFG CORPORATION
28	05452	RAYTHEON CO MSL SYS DIV
29	05716	RAYTHEON CO
30	05716	RAYTHEON CO.
31	05869	HUGHES AIRCRAFT COMPANY
32	06401	SPERRY CORP
33	06509	ENGINEERED MAGNETICS INC.
34	06581	ALLAN AIRCRAFT SUPPLY CO
35	06721	MIDLAND BRAKE INC
36	06983	ZAGORA GEAR PRODUCTS INC.
37	07076	RCA CORP GOV SYS DIV AUTOMATED SYS
38	07076	RCA CORPORATION
39	07145	TIMES FIBER COMMUNICATIONS CO.
40	07178	CONSOLIDATED MACH & TOOL CORP
41	07187	HONEYWELL INC
42	07187	SPERRY CORP
43	07187	SPERRY CORP SPERRY DEFENSE SYS DIV
44	07187	SPERRY CORPORATION
45	07239	BIDDLE INSTRUMENTS
46	07486	SYERDRUP TECHNOLOGY INC.
47	07862	ELECTRO CIRCUITS INC
48	08040	OMNI-WAVE ELECTRONICS CORP
49	08638	HONEYWELL INC
50	08638	HONEYWELL INCORPORATED
51	08719	JAY EL PRODUCTS INC
52	08903	BOEING AEROSPACE COMPANY
53	08996	GENERAL PNEUMATICS CORP
54	09036	SON FARREL INC
55	09036	SON FARREL, INC
56	09036	SONFARREL INC
57	09036	SONFARREL, INC.
58	09087	INFODEX, INC.
59	09205	FORD AEROSPACE & COMM CORP
60	09205	FORD AEROSPACE AND COMM CORP

FIGURE 25

4.0 Summary

4.1 A valid copy of the DRS database located on the IMD IBM mainframe must be downloaded to the CFO Unisys 5000/80 on a weekly basis.

4.2 An update procedure needs to be developed so that the entire database does not have to be downloaded each week.

5.0 Conclusion

With the exception of three sort routines and templating of QM to originator correspondence, the tasks required in the SOW have been met.

6.0 Recommendations

6.1 A closer working relationship needs to be developed between IMD and the CFO and its support contractor. Direct discussions between IMD and the support contractor are required to discuss technical matters pertaining to the downloading of the DRS and the operation of the Empress database systems. IMD and the support contractor both have the mission of meeting operational requirements of the mutual customer, PAD, and a partnership needs to be established to meet these requirements.

6.2 Plans should be made to develop a translator program that updates both the S2K and CFO databases in one data entry session by the CFO computer operators. One-time entry reduces the possibility of keystroke errors.

6.3 Only DRS data for the last three years is currently being dumped from the S2K system. It is recommended that all DRS data be dumped from the S2K system and loaded into the PAD2 database. Subsequently, only data updates should be downloaded from the S2K system.

6.4 The present DRS database update process involves downloading S2K DRS data from the IMD computer onto a magnetic tape which is then loaded into the DRS tables created by (and "owned" by) IMD personnel on the PAD2 computer. In order to provide "ownership" rights, controls and privileges to the PAD DRS database administrator, data from the IMD-created tables are again downloaded and then uploaded to tables created and "owned" by CFO. This results in having two sets of the same DRS data installed on the PAD2 computer. It is recommended that one set of the DRS data be eliminated and have the IMD database, NDRS, be given to CFO with all privileges granted.

6.5 It is strongly urged that special effort be taken by PAD management to expedite the linking of the two PAD Unisys 5000 computers to each other and to the MICOM Unisys network. This would greatly facilitate the transfer of PAD-related data and would allow direct and speedy PAD access to programs and databases developed or being developed by other MICOM elements (especially program management offices).

6.6 It is recommended that the QDR alerting system be expanded to include tracking of requests for support and subsequent follow-up letters by the action officers. This expansion effort should also investigate the feasibility of having the computer search the DRS database tables and prepare form letter follow up requests to the support points, with the form letters amended by the

action officer to suit the specifics of the case. The appropriate action process codes and dates could then be inserted automatically into the database tables.

6.7 The QDR alerting system should also be augmented by having appropriate menu-driven query routines developed to assist action officers and others in rapid retrieval of selected data and in responding to ad-hoc query situations. In particular, mvision screens should be developed for improved data entry procedures.

6.8 A special effort should be made to include all Warranty Claim Actions (WCAs) and Equipment Improvement Reports (EIRs) in the PAD DRS database. It is quite probable that this would require some additional special tables to contain related information not presently collected in the S2K system. Especially, the new Critical Safety Item database now being developed for PAD should be related to the present DRS database so as to assure that QDRs on these items will automatically be treated as Category I QDRs.

6.9 A survey should be made of PAD personnel to determine their needs regarding specific types of data which is (or could be) contained in the DRS database tables. For example, special reports or menu-driven query screens could be developed for providing analysis of deficiencies by contractor, by weapon system, by type of hardware (missile, launcher, radar, etc.), by failure modes or by operating time at failure. Statistical analysis of root cause of failures could be developed if the database contains sufficient suitable data.

APPENDIX A

Task Plan and Milestone Chart, 17 March 1989

HILTON SYSTEMS, INC. REPORT NO.: HSI/89-0013-01

ESTABLISHMENT OF DEFICIENCY REPORT MANAGEMENT SYSTEM

TASK PLAN AND MILESTONE CHART

14 MARCH 1989

Prepared For:

Commander
U.S. Army Missile Command
ATTN: Mr. Thomas L. Moore
Product Assurance Directorate
AMSMI-QA-CF
Redstone Arsenal, AL 35898-5290

PREPARED BY: M. Wayne Miller, Jr.

CONTRACT: DAAH01-89-D-0013/0005

CLASSIFICATION AND CONTENT APPROVED BY:
Harold R. Bright
Director, Product Assurance

DATA ITEM: A002 (SOW 5.1)

1.0 Purpose

The purpose of this plan is to present the contractors approach for meeting the requirements of the statement of work for delivery order number 0005 for contract number DAAH01-89-D-0013.

The primary objective of this statement of work is to have the contractor develop procedures for assigning management goals for accomplishment of several activities and for measuring the progress on meeting those goals over monthly, quarterly and annual time periods. The contractor is also directed to improve procedures for summarizing in monthly reports the deficiency report processing activities and procedures for periodic update.

2.0 Discussion

The tasks required to accomplish this delivery order statement of work are depicted on the attached milestone chart. What follows is a discussion of those tasks that are not self-explanatory.

2.1 Review Empress 2

The purpose of this task is to enable the contractor personnel to become familiar with the database as it resides on the CFO Sperry 5000/80. This includes construct, data elements available, query requirements, etc. Although six weeks is scheduled for the task, it is looked upon as being continuous throughout the length of the task. The six week period is a concentrated effort to meet summary report requirements.

2.2 Investigate Graphics Capabilities

This task will determine the organic graphic capabilities that exist within CFO/PAD taking into consideration future capabilities that have been approved for procurement but not yet on hand. The available capabilities will be employed to develop graphic presentations of data for briefing management officials, Project Managers, Program Executive Officers and higher headquarters.

2.3 Match S2K to Empress 2

This long term task is designed to ensure that no required data elements are lost in the transfer of the Deficiency Reporting System data from the mainframe to the PAD Sperry 5000/80. This is planned for the sole purpose of providing the best available data to the user decision makers.

2.4 Prepare Prototype Summary Report

The reason for the "prototype" report is because the contractor operates under the premise of; show the user a possible solution, let him/her operate it and tell what he/she likes, dislikes, wants added or taken away; rewrite the solution and incorporate as many of the user desires as technically possible. This procedure has proven itself time and time again

as the best way to provide a usable product. Iteration and review is limited by time and systems available.

2.5 Prepare Sort Routines

These routines will allow the user to look at summaries based on weapon systems, subsystem, action officer, contractor, origin of deficiency report and by type of report. The desire is that these different sorts will be available to the user from an on screen menu.

2.6 Prepare Outgoing Letters and Reports

This involves the preparation of: acknowledgment letters; deficiency investigation letters; exhibit shipping instructions; interim and final replies to deficiency report originator; investigative follow-up letters; advisory notices to other government elements and special categories of deficiency reports. It is also desirable that these will be available from a menu with some fill in the blanks data entries.

3.0 Summary

In the accomplishment of this task the contractor plans to prepare possible solutions and iterate changes through user hands-on reviews. This procedure will be employed for all requirements mentioned above. The contractor wants to provide access to all the products with the minimal input by the user preferably through selection menus.

4.0 Attachments

One, the task milestone chart.

5.0 References

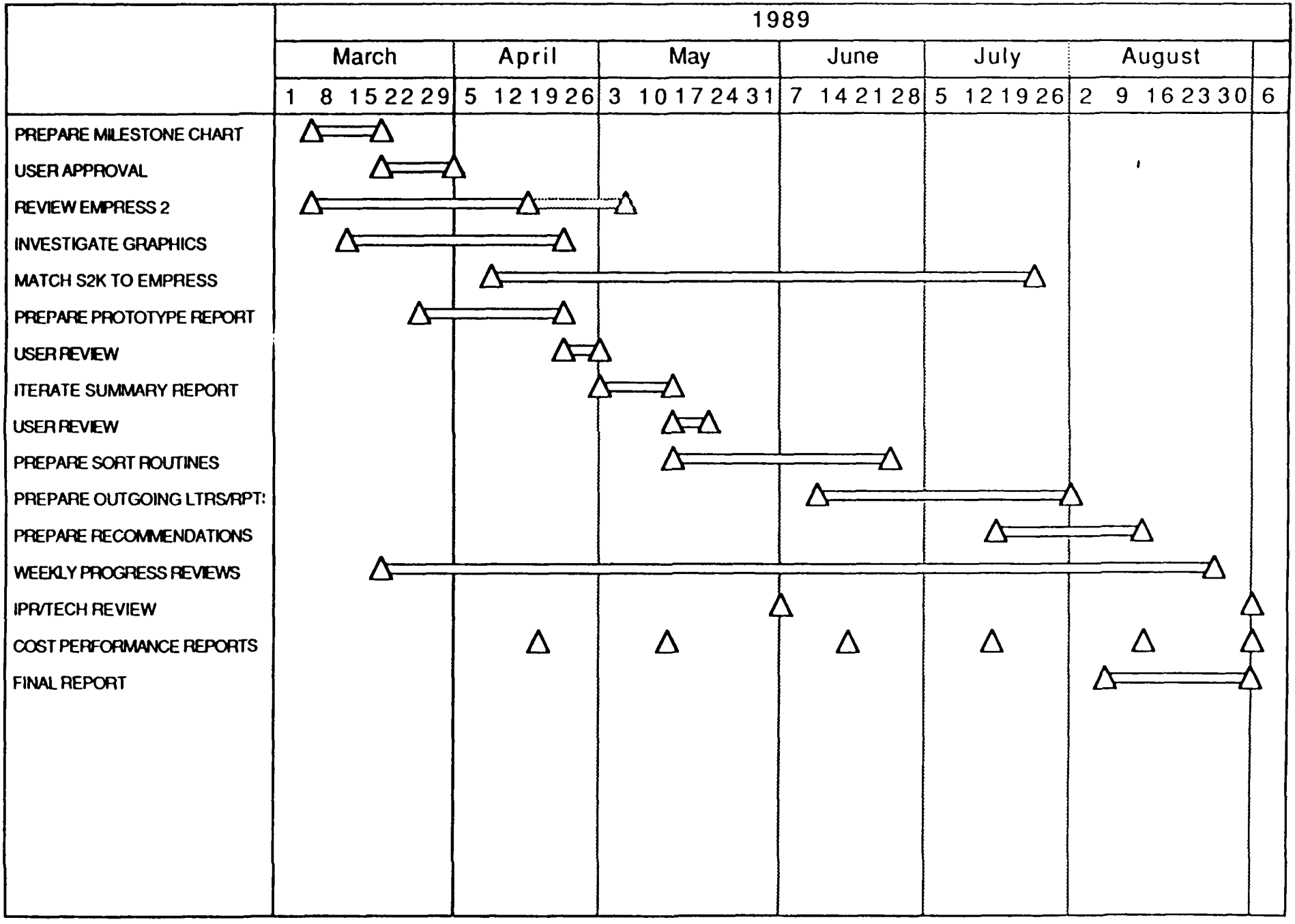
None.

6.0 Appendix

None.

TASK MILESTONE CHART

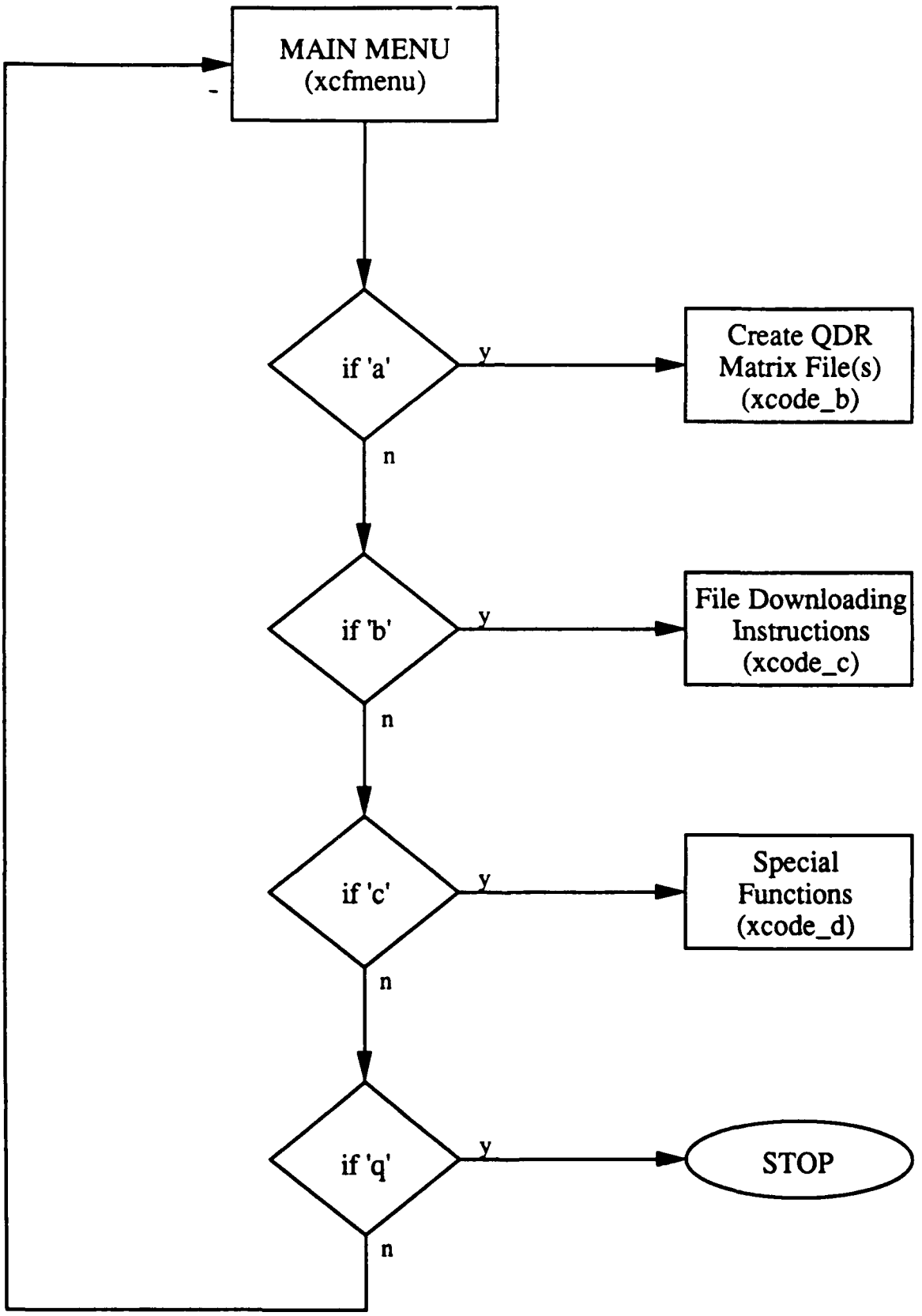
ESTABLISHMENT OF DEFICIENCY REPORT MANAGEMENT SYSTEM



APPENDIX B

Main Menu Source Code

CUSTOMER FEEDBACK OFFICE QDR INFORMATION SYSTEM



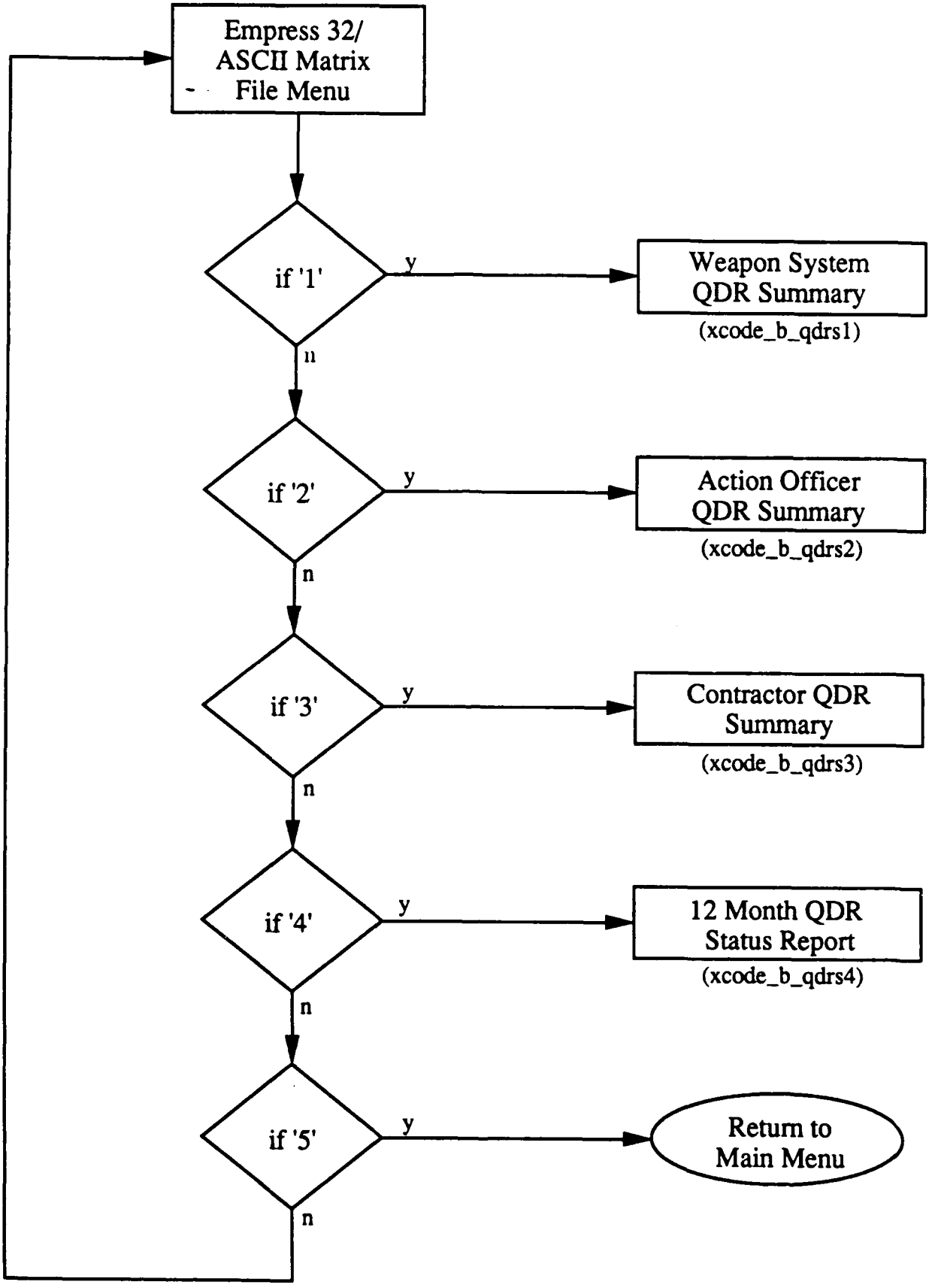
```
 1  clear
 2  input=
 3  while (:)
 4  dta=`date | cut -c1-3`
 5  dtc=`date | cut -c5-10`
 6  dtb=`date | cut -c25-28`
 7  tt=`date | cut -c12-16`
 8  echo;
 9  echo "\t$dta", "$dtc", "$dtb"
10  echo "\t$tt"
11  echo "\t*************************************************************
12  echo "\t*                                                           *
13  echo "\t*       Customer Feedback Office QDR Information System     *
14  echo "\t* ___________________________________________________________ *
15  echo "\t*                                                           *
16  echo "\t*                                                           *
17  echo "\t*                                              Code         *
18  echo "\t*                                                           *
19  echo "\t*           Create QDR Matrix File(s)           a           *
20  echo "\t*           File Downloading Instructions       b           *
21  echo "\t*           Special Functions                   c           *
22  echo "\t*           Quit                                q           *
23  echo "\t*                                                           *
24  echo "\t*                                                           *
25  echo "\t*                                                           *
26  echo "\t*                                                           *
27  echo "\t*************************************************************
28  echo;
29  echo "\tEnter Code : \c"
30  read input

31  do
32  case $input in

33  [Aa])
34  xcode_b;
35  ;;

36  [Bb])
37  xcode_c;
38  clear;
39  ;;

40  [Cc])
41  xcode_d;
42  ;;

43  [Qq])
44  clear
45  echo;
46  echo;
47  echo;
48  echo;
49  echo;
50  echo;
51  echo;
52  echo;
53  echo "\t\t\t   *** Exit From Menu ***";
54  sleep 2;
55  clear
```

```
56  break
57  ;;

58  *)
59  clear;
60  echo;
61  echo;
62  echo "\t\t      You Have Made An Invalid Selection !";
63  echo;
64  echo;
65  echo "\t\t              Please Try Again.";
66  sleep 4;
67  clear
68  ;;

69  esac
70  done
```

APPENDIX C

Empress 32/ASCII Matrix File Menu Source Code

CREATE QDR MATRIX FILE(S) (xcode_b)



```
sel=
while (:)
clear
echo;
echo;
echo;
echo "\t\t\tEMPRESS 32/ASCII MATRIX FILE MENU"
echo "\t_____________________________________________________________"
echo;
echo;
echo "\t\t\t1-  Weapon System QDR Summary"
echo "\t\t\t2-  Action Officer QDR Summary"
echo "\t\t\t3-  Contractor QDR Summary"
echo "\t\t\t4-  12 Month QDR Status Report"
echo "\t\t\t5-  Main Menu"
echo;
echo;
echo "\tEnter Choice : \c"
read choice

do
case "$choice" in
1)
clear;
echo;
echo "\t\t\tWeapon System QDR Summary Matrix"
echo;
echo;
echo "\tEnter Matrix Filename : \c"
read file1
if test $file1 = q
then clear
     break
fi
echo "\tEnter Beginning Date (mmddyyyy) : \c";
read date1;
echo $date1 >datef1
cut -c1-4 datef1 >datef1a
cut -c5-8 datef1 >datef1b
paste datef1b datef1a >date1ba
sed 's, ,,g' date1ba >date1nul
date1c=`cat date1nul`
echo "\tEnter Ending Date (mmddyyyy)    : \c";
read date2;
echo $date2 >datef2
cut -c1-4 datef2 >datef2a
cut -c5-8 datef2 >datef2b
paste datef2b datef2a >date2ba
sed 's, ,,g' date2ba >date2nul
date2c=`cat date2nul`
rm date1* date2* datef*
clear;
tt=`date | cut -c12-16`
echo "$tt";
echo;
echo "\t     Processing Weapon System QDR Data, Please Wait ... \c";
xcode_b_qdrs1 $date1c $date2c $file1
#zqdr $date1 $date2;
clear;
echo;
```

```
60  info=`l $file1 | cut -c16-68`
61  echo;
62  echo "\t    $tt    $info";
63  xcursor 23 27
64  echo "press <Enter> to continue";
65  read pause;
66  clear;
67  cat $file1
68  read pause;
69  ;;

70  2)
71  clear;
72  echo;
73  echo "\t\t\tAction Officer QDR Summary Matrix";
74  echo;
75  echo;
76  echo "\tEnter Matrix Filename : \c";
77  read file1
78  if test $file1 = q
79  then clear
80       break
81  fi
82  echo "\tEnter Beginning Date (mmddyyyy) : \c";
83  read date1;
84  echo $date1 >datef1
85  cut -c1-4 datef1 >datef1a
86  cut -c5-8 datef1 >datef1b
87  paste datef1b datef1a >date1ba
88  sed 's, ,,g' date1ba >date1nul
89  date1c=`cat date1nul`
90  echo "\tEnter Ending Date (mmddyyyy)    : \c";
91  read date2;
92  echo $date2 >datef2
93  cut -c1-4 datef2 >datef2a
94  cut -c5-8 datef2 >datef2b
95  paste datef2b datef2a >date2ba
96  sed 's, ,,g' date2ba >date2nul
97  date2c=`cat date2nul`
98  rm date1* date2* datef*
99  clear
100 tt=`date | cut -c12-16`
101 echo "$tt";
102 echo;
103 echo "\t    Processing Action Officer QDR Data, Please Wait ... \c";
104 xcode_b_qdrs2 $date1c $date2c $file1
105 clear;
106 echo;
107 info=`l $file1 | cut -c16-68`
108 echo;
109 echo "\t    $tt    $info";
110 xcursor 23 27
111 echo "press <Enter> to continue";
112 read pause;
113 clear;
114 cat $file1
115 read pause;
116 ;;

117 3)
```

```
118  clear;
119  echo
120  echo "\t\t\tContractor QDR Summary Matrix"
121  echo
122  echo
123  echo "\tEnter Matrix Filename : \c"
124  read file1
125  if test $file1 = q
126  then clear
127       break
128  fi
129  echo "\tEnter Beginning Date (mmddyyyy) : \c"
130  read date1
131  echo $date1 >datef1
132  cut -c1-4 datef1 >datef1a
133  cut -c5-8 datef1 >datef1b
134  paste datef1b datef1a >date1ba
135  sed 's, ,,g' date1ba >date1nul
136  date1c=`cat date1nul`
137  echo "\tEnter Ending Date (mmddyyyy)    : \c"
138  read date2
139  echo $date2 >datef2
140  cut -c1-4 datef2 >datef2a
141  cut -c5-8 datef2 >datef2b
142  paste datef2b datef2a >date2ba
143  sed 's, ,,g' date2ba >date2nul
144  date2c=`cat date2nul`
145  rm date1* date2* datef*
146  clear
147  tt=`date | cut -c12-16`
148  echo "$tt"
149  echo
150  echo "\t    Processing Contractor QDR Data, Please Wait ... \c"
151  xcode_b_qdrs3 $date1c $date2c $file1
152  clear
153  echo
154  info=`l 31.$file1 | cut -c16-68`
155  info2=`l 32.$file1 | cut -c16-68`
156  info3=`l 33.$file1 | cut -c16-68`
157  info4=`l 34.$file1 | cut -c16-68`
158  info5=`l 35.$file1 | cut -c16-68`
159  info6=`l 36.$file1 | cut -c16-68`
160  info7=`l 37.$file1 | cut -c16-68`
161  info8=`l 38.$file1 | cut -c16-68`
162  info9=`l 39.$file1 | cut -c16-68`
163  echo
164  echo "\t    $tt    $info"
165  echo "\t             $info2"
166  echo "\t             $info3"
167  echo "\t             $info4"
168  echo "\t             $info5"
169  echo "\t             $info6"
170  echo "\t             $info7"
171  echo "\t             $info8"
172  echo "\t             $info9"
173  xcursor 23 27
174  echo "press <Enter> to continue"
175  read pause
176  clear
177  ;;
```

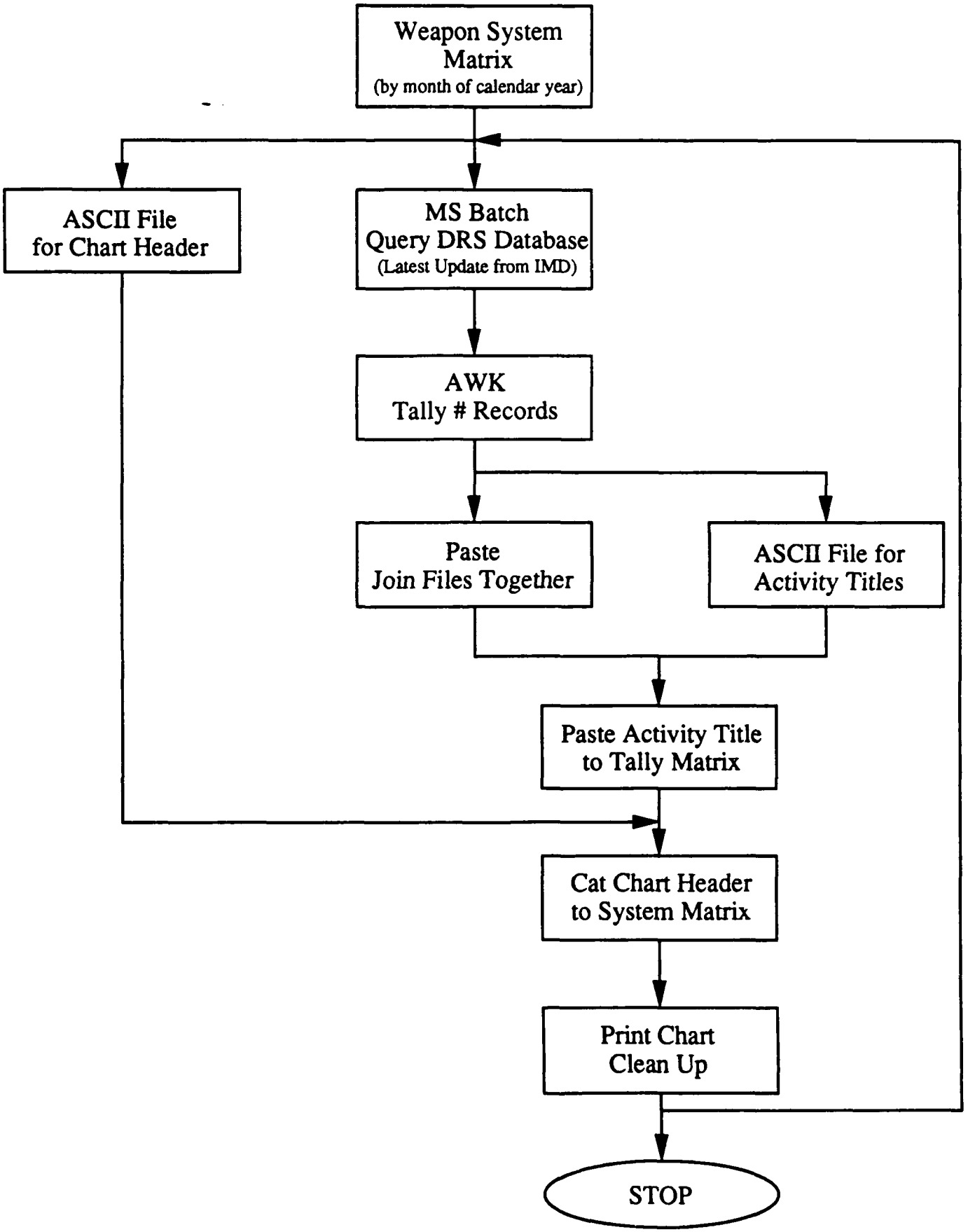
```
178  4)
179  clear
180  echo
181  echo "\t\t\tQDR Status Report Matrix"
182  echo
183  echo
184  echo "\tEnter Matrix Filename : \c"
185  read file1
186  if test $file1 = q
187  then clear
188       break
189  fi
190  echo "\tEnter Beginning Date (mmddyyyy) : \c"
191  read date1
192  echo $date1 >datef1
193  cut -c1-4 datef1 >datef1a
194  cut -c5-8 datef1 >datef1b
195  paste datef1b datef1a >date1ba
196  sed 's, ,,g' date1ba >date1nul
197  date1c=`cat date1nul`
198  echo "\tEnter Ending Date (mmddyyyy)    : \c"
199  read date2
200  echo $date2 >datef2
201  cut -c1-4 datef2 >datef2a
202  cut -c5-8 datef2 >datef2b
203  paste datef2b datef2a >date2ba
204  sed 's, ,,g' date2ba >date2nul
205  date2c=`cat date2nul`
206  cut -c1-2 datef1 >zmo2
207  zmo=`cat zmo2`
208  zyr=`cat datef1b`
209  if test $zmo = 01
210  then sumpd=JAN
211  fi
212  if test $zmo = 02
213  then sumpd=FEB
214  fi
215  if test $zmo = 03
216  then sumpd=MAR
217  fi
218  if test $zmo = 04
219  then sumpd=APR
220  fi
221  if test $zmo = 05
222  then sumpd=MAY
223  fi
224  if test $zmo = 06
225  then sumpd=JUN
226  fi
227  if test $zmo = 07
228  then sumpd=JUL
229  fi
230  if test $zmo = 08
231  then sumpd=AUG
232  fi
233  if test $zmo = 09
234  then sumpd=SEP
235  fi
236  if test $zmo = 10
```

```
237  then sumpd=OCT
238  fi
239  if test $zmo = 11
240  then sumpd=NOV
241  fi
242  if test $zmo = 12
243  then sumpd=DEC
244  fi
245  rm date1* date2* datef* zmo2
246  clear
247  tt=`date | cut -c12-16`
248  echo "$tt"
249  echo
250  echo "\t    Processing QDR Status Report Data, Please Wait ... \c"
251  #echo "date1c : "$date1c" date2c : "$date2c" sumpd : "$sumpd" zyr : "$zy
252  xcode_b_qdrs4 $date1c $date2c $sumpd $zyr $file1
253  clear
254  echo
255  info=`l $file1 | cut -c16-68`
256  echo
257  echo "\t    $tt    $info"
258  xcursor 23 27
259  echo "press <Enter> to continue"
260  read pause
261  clear
262  cat $file1
263  read pause
264  ;;

265  5)
266  clear;
267  break;
268  ;;

269  esac
270  done
```

APPENDIX D

Weapon System QDR Summary Source Code

WEAPON SYSTEM QDR SUMMARY



```
 1  msbatch /usr2/acct/tnewby/drs/ndrs << }
 2  set MSVALSEP to '       ';

 3  select wpn_sys_cd from s2krg0 dump into OBOP where docu_mgt_no match "P*

 4  select wpn_sys_cd from s2krg0 dump into RCDP where docu_mgt_no match "P*

 5  select wpn_sys_cd from s2krg0 dump into CDRP where docu_mgt_no match "P*

 6  select wpn_sys_cd from s2krg0 dump into TDRP where docu_mgt_no match "P*

 7  select wpn_sys_cd from s2krg0 dump into OEOP where docu_mgt_no match "P*

 8  select wpn_sys_cd from s2krg0 dump into O1_120 where docu_mgt_no match "

 9  select wpn_sys_cd from s2krg0 dump into O120_180 where docu_mgt_no match
10  and act_pt != "QACF";

11  select wpn_sys_cd from s2krg0 dump into OV180 where docu_mgt_no match "P

12  select wpn_sys_cd from s2krg0 dump into OV180C where docu_mgt_no match "
13  }

14  awk -f 0419a OBOP > OBOP.tal
15  awk -f 0419b RCDP > RCDP.tal
16  awk -f 0419b CDRP > CDRP.tal
17  awk -f 0419b TDRP > TRDP.tal
18  awk -f 0419b OEOP > OEOP.tal
19  awk -f 0419b O1_120 > O1_120.tal
20  awk -f 0419b O120_180 > O120_180.tal
21  awk -f 0419b OV180 > OV180.tal
22  awk -f 0419b OV180C > OV180C.tal

23  paste OBOP.tal RCDP.tal > chart
24  paste chart CDRP.tal > chart1
25  paste chart1 TRDP.tal > chart2
26  paste chart2 OEOP.tal > chart3
27  paste chart3 O1_120.tal > chart4
28  paste chart4 O120_180.tal > chart5
29  paste chart5 OV180.tal > chart6
30  paste chart6 OV180C.tal > matrix

31  dt=$2
32  echo $dt >zdt2
33  cut -c1-4 zdt2 >zyr
34  cut -c3-4 zdt2 >zyrx
35  cut -c5-6 zdt2 >zmo
36  cut -c7-8 zdt2 >zda
37  slsh="/"
38  echo $slsh >zslsh
39  paste zmo zslsh >z2mo
40  paste zda zslsh >z2da
41  paste z2mo z2da zyr >zdate
42  sed 's, ,,g' zdate >z2date
43  heading z2date >zheading
44  cat zheading matrix >$3

45  awk -f xc_qdrs1_awk OBOP >OBOP.tal
46  awk -f xc_qdrs1_awk RCDP >RCDP.tal
47  awk -f xc_qdrs1_awk CDRP >CDRP.tal
```

```
48  awk -f xc_qdrs1_awk TDRP >TDRP.tal
49  awk -f xc_qdrs1_awk OEOP >OEOP.tal
50  awk -f xc_qdrs1_awk O1_120 >O1_120.tal
51  awk -f xc_qdrs1_awk O120_180 >O120_180.tal
52  awk -f xc_qdrs1_awk OV180 >OV180.tal
53  awk -f xc_qdrs1_awk OV180C >OV180C.tal
54  paste OBOP.tal RCDP.tal >chart
55  paste chart CDRP.tal >chart1
56  paste chart1 TDRP.tal >chart2
57  paste chart2 OEOP.tal >chart3
58  paste chart3 O1_120.tal >chart4
59  paste chart4 O120_180.tal >chart5
60  paste chart5 OV180.tal >chart6
61  zmo2=`cat zmo`
62  zyr2=`cat zyrx`
63  paste chart6 OV180C.tal >$zmo2$zyr2.mac

64  rm zdt2 zmo zda zyr zyrx z2mo z2da zslsh zdate z2date zheading
65  rm chart* *.tal CDRP O120_180 O1_120 OBOP OEOP OV180 OV180C RCDP TDRP
```

```
BEGIN {
        OFS = "\t"
        bx=dc=a2c=de=ds=dh=dj=dr=d6=d7=ef=en=ep=ey=e2=e3=e5=e7=0
        fa=fc=fg=fp=fq=fr=f3=f4=mw=h4=oo=td=xx=0
        a22=a23=a25=a2f=a2g=a2h=a2n=a2p=a2r=a2s=0
}
$1 == "BX" {bx++}
$1 == "DC" {dc++}
$1 == "2C" {a2c++}
$1 == "DE" {de++}
$1 == "DS" {ds++}
$1 == "DH" {dh++}
$1 == "DJ" {dj++}
$1 == "DR" {dr++}
$1 == "D6" {d6++}
$1 == "D7" {d7++}
$1 == "EF" {ef++}
$1 == "EN" {en++}
$1 == "EP" {ep++}
$1 == "EY" {ey++}
$1 == "E2" {e2++}
$1 == "E3" {e3++}
$1 == "E5" {e5++}
$1 == "E7" {e7++}
$1 == "FA" {fa++}
$1 == "FC" {fc++}
$1 == "FG" {fg++}
$1 == "FP" {fp++}
$1 == "FQ" {fq++}
$1 == "FR" {fr++}
$1 == "F3" {f3++}
$1 == "F4" {f4++}
$1 == "MW" {mw++}
$1 == "XX" {xx++}
$1 == "HD" {hd++}
$1 == "TD" {td++}
$1 == "00" {oo++}
$1 == "22" {a22++}
$1 == "23" {a23++}
$1 == "25" {a25++}
$1 == "2F" {a2f++}
$1 == "2G" {a2g++}
$1 == "2H" {a2h++}
$1 == "2N" {a2n++}
$1 == "2P" {a2p++}
$1 == "2R" {a2r++}
$1 == "2S" {a2s++}
END { tot=bx+dc+a2c+de+ds+a2s+dh+a2h+dj+dr+a2r+d6+d7+ef+a2f+en+a2n+ep+a2
      tot1=ey+e2+a22+e3+a23+e5+a25+e7+fa+fc+fg+a2g+fp+fq+fr+f3+f4+mw
      tot2=td+h4+oo+xx
      total=tot+tot1+tot2
      print " ", "OH-58D      ", bx
      print " ", "CHAPARRAL   ", (dc + a2c)
      print " ", "HAWK        ", (de + ds + a2s)
      print " ", "TARGETS     ", (dh + a2h)
      print " ", "REDEYE      ", dj
      print " ", "STINGER     ", (dr + a2r)
      print " ", "PATRIOT     ", d6
      print " ", "FAAR        ", d7
      print " ", "MLRS        ", (ef + a2f)
```

```
61          print " ", "LANCE       ", (en + a2n)
62          print " ", "HELLFIRE    ", (ep + a2p)
63          print " ", "LCSS        ", ey
64          print " ", "TOW         ", (e2 + a22)
65          print " ", "PERSHING II", (e3 + a23)
66          print " ", "DRAGON      ", (e5 + a25)
67          print " ", "TOW BRADLEY", e7
68          print " ", "GLLD        ", fa
69          print " ", "MULE        ", fc
70          print " ", "THERM IMAGE", (fg + a2g)
71          print " ", "TADS        ", fp
72          print " ", "CALIBRATION", fq
73          print " ", "APACHE      ", fr
74          print " ", "AN/TSQ-73   ", f3
75          print " ", "TOW COBRA   ", f4
76          print " ", "FIST-V      ", mw
77          print " ", "UNKNOWN     ", (td + h4 + oo + xx)
78          print " ", "==============", "======"
79          print " ", "TOTAL       ", total
80  }
```

```
BEGIN {
      OFS = "\t"
      bx=dc=a2c=de=ds=dh=dj=dr=d6=d7=ef=en=ep=ey=e2=e3=e5=e7=0
      fa=fc=fg=fp=fq=fr=f3=f4=mw=h4=oo=td=xx=0
      a22=a23=a25=a2f=a2g=a2h=a2n=a2p=a2r=a2s=0
}
$1 == "BX" {bx++}
$1 == "DC" {dc++}
$1 == "2C" {a2c++}
$1 == "DE" {de++}
$1 == "DS" {ds++}
$1 == "DH" {dh++}
$1 == "DJ" {dj++}
$1 == "DR" {dr++}
$1 == "D6" {d6++}
$1 == "D7" {d7++}
$1 == "EF" {ef++}
$1 == "EN" {en++}
$1 == "EP" {ep++}
$1 == "EY" {ey++}
$1 == "E2" {e2++}
$1 == "E3" {e3++}
$1 == "E5" {e5++}
$1 == "E7" {e7++}
$1 == "FA" {fa++}
$1 == "FC" {fc++}
$1 == "FG" {fg++}
$1 == "FP" {fp++}
$1 == "FQ" {fq++}
$1 == "FR" {fr++}
$1 == "F3" {f3++}
$1 == "F4" {f4++}
$1 == "MW" {mw++}
$1 == "XX" {xx++}
$1 == "HD" {hd++}
$1 == "TD" {td++}
$1 == "00" {oo++}
$1 == "22" {a22++}
$1 == "23" {a23++}
$1 == "25" {a25++}
$1 == "2F" {a2f++}
$1 == "2G" {a2g++}
$1 == "2H" {a2h++}
$1 == "2N" {a2n++}
$1 == "2P" {a2p++}
$1 == "2R" {a2r++}
$1 == "2S" {a2s++}
END { tot=bx+dc+a2c+de+ds+a2s+dh+a2h+dj+dr+a2r+d6+d7+ef+a2f+en+a2n+ep+a2
      tot1=ey+e2+a22+e3+a23+e5+a25+e7+fa+fc+fg+a2g+fp+fq+fr+f3+f4+mw
      tot2=td+h4+oo+xx
      total=tot+tot1+tot2
      print bx
      print (dc + a2c)
      print (de + ds + a2s)
      print (dh + a2h)
      print dj
      print (dr + a2r)
      print d6
      print d7
      print (ef + a2f)
```

```
61          print (en + a2n)
62          print (ep + a2p)
63          print ey
64          print (e2 + a22)
65          print (e3 + a23)
66          print (e5 + a25)
67          print e7
68          print fa
69          print fc
70          print (fg + a2g)
71          print fp
72          print fq
73          print fr
74          print f3
75          print f4
76          print mw
77          print (td + h4 + oo + xx)
78          print "======"
79          print total
80  }
```

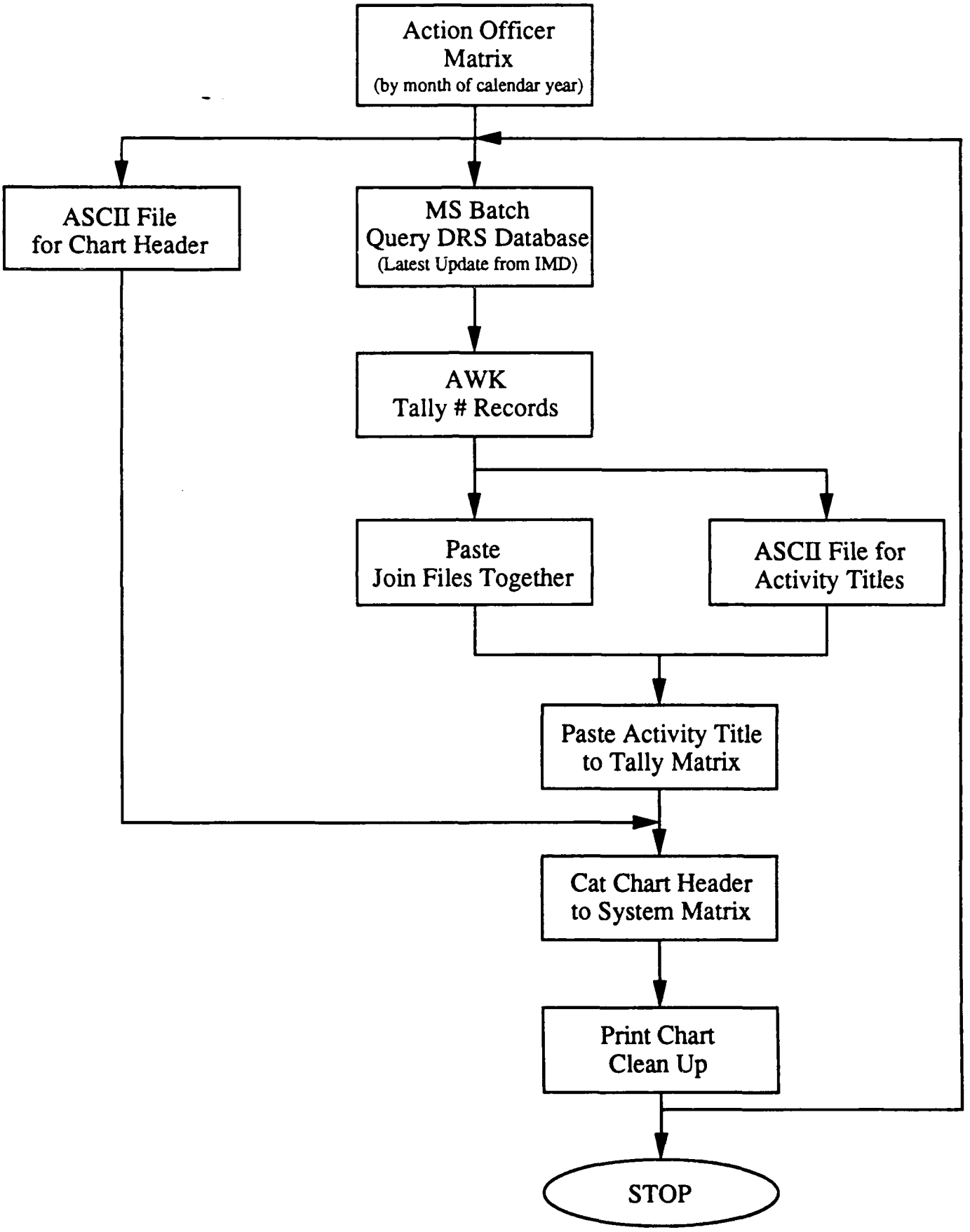
```
 1  dta=`date | cut -c5-7`
 2  dtb=`date | cut -c25-28`
 3  dtc=`date | cut -c9-10`
 4  #sumpd=`cat $1`
 5  cut -c1-2 $1 >zmo2
 6  cut -c7-10 $1 >zyr2
 7  zmo=`cat zmo2`
 8  zyr=`cat zyr2`
 9  if test $zmo = 01
10  then sumpd=JANUARY
11       sp=__
12  fi
13  if test $zmo = 02
14  then sumpd=FEBRUARY
15       sp=_
16  fi
17  if test $zmo = 03
18  then sumpd=MARCH
19       sp=____
20  fi
21  if test $zmo = 04
22  then sumpd=APRIL
23       sp=____
24  fi
25  if test $zmo = 05
26  then sumpd=MAY
27       sp=_____
28  fi
29  if test $zmo = 06
30  then sumpd=JUNE
31       sp=_____
32  fi
33  if test $zmo = 07
34  then sumpd=JULY
35       sp=_____
36  fi
37  if test $zmo = 08
38  then sumpd=AUGUST
39       sp=___
40  fi
41  if test $zmo = 09
42  then sumpd=SEPTEMBER
43  fi
44  if test $zmo = 10
45  then sumpd=OCTOBER
46       sp=__
47  fi
48  if test $zmo = 11
49  then sumpd=NOVEMBER
50       sp=_
51  fi
52  if test $zmo = 12
53  then sumpd=DECEMBER
54       sp=_
55  fi
56  rm zmo2 zyr2

57  echo;
58  echo "        QUALITY DEFICIENCY REPORT SUMMARY"
59  echo "                    AMSMI-QA-CF                    SUMMARY PERIOD :
```

```
60  echo "
61  echo;
62  echo;
63  echo "        WEAPON              OBOP     RCVD     CLOSED   TRANSF   OEOP     1-
64  echo "        SYSTEM                                                           DA
65  echo "        ==============      ======   ======   ======   ======   ======   ==
```

APPENDIX E

Action Officer QDR Summary Source Code

ACTION OFFICER QDR SUMMARY



```
 1  msbatch /usr2/acct/tnewby/drs/ndrs << }
 2  set MSVALSEP to '       ';

 3  select act_off_cd from s2krg0 dump into OBOP where docu_mgt_no match "P*

 4  select act_off_cd from s2krg0 dump into RCDP where docu_mgt_no match "P*

 5  select act_off_cd from s2krg0 dump into CDRP where docu_mgt_no match "P*

 6  select act_off_cd from s2krg0 dump into TDRP where docu_mgt_no match "P*

 7  select act_off_cd from s2krg0 dump into OEOP where docu_mgt_no match "P*

 8  select act_off_cd from s2krg0 dump into O1_120 where docu_mgt_no match "

 9  select act_off_cd from s2krg0 dump into O120_180 where docu_mgt_no match
10  and act_pt != "QACF";

11  select act_off_cd from s2krg0 dump into OV180 where docu_mgt_no match "P

12  select act_off_cd from s2krg0 dump into OV180C where docu_mgt_no match "
13  }

14  awk -f 0419a2 OBOP > OBOP.tal
15  awk -f 0419b2 RCDP > RCDP.tal
16  awk -f 0419b2 CDRP > CDRP.tal
17  awk -f 0419b2 TDRP > TRDP.tal
18  awk -f 0419b2 OEOP > OEOP.tal
19  awk -f 0419b2 O1_120 > O1_120.tal
20  awk -f 0419b2 O120_180 > O120_180.tal
21  awk -f 0419b2 OV180 > OV180.tal
22  awk -f 0419b2 OV180C > OV180C.tal

23  paste OBOP.tal RCDP.tal > chart
24  paste chart CDRP.tal > chart1
25  paste chart1 TRDP.tal > chart2
26  paste chart2 OEOP.tal > chart3
27  paste chart3 O1_120.tal > chart4
28  paste chart4 O120_180.tal > chart5
29  paste chart5 OV180.tal > chart6
30  paste chart6 OV180C.tal > matrix

31  dt=$2
32  echo $dt >zdt2
33  cut -c1-4 zdt2 >zyr
34  cut -c5-6 zdt2 >zmo
35  cut -c7-8 zdt2 >zda
36  slsh="/"
37  echo $slsh >zslsh
38  paste zmo zslsh >z2mo
39  paste zda zslsh >z2da
40  paste z2mo z2da zyr >zdate
41  sed 's, ,,g' zdate >z2date
42  heading2 z2date >zheading
43  cat zheading matrix > $3
44  rm zdt2 zmo zda zyr z2mo z2da zslsh zdate z2date zheading

45  rm chart* *.tal CDRP O120_180 O1_120 OBOP OEOP OV180 OV180C RCDP TDRP
```

```
 1  BEGIN {
 2        OFS = "\t"
 3        lmb=gsr=sjs=jc=bj=rt=ea=mj=cfo=0
 4  }
 5  $1 == "LMB" {lmb++}
 6  $1 == "GSR" {gsr++}
 7  $1 == "SJS" {sjs++}
 8  $1 == "JC"  {jc++}
 9  $1 == "BJ"  {bj++}
10  $1 == "RT"  {rt++}
11  $1 == "EA"  {ea++}
12  $1 == "MJ"  {mj++}
13  $1 == "CFO" {cfo++}

14  END { total=lmb+gsr+sjs+jc+bj+rt+ea+mj+cfo

15        print " ", "L. Blanton   ", lmb
16        print " ", "G. Ross      ", gsr
17        print " ", "S. Scott     ", sjs
18        print " ", "J. Cooper    ", jc
19        print " ", "B. James     ", bj
20        print " ", "R. Thomas    ", rt
21        print " ", "E. Ashworth  ", ea
22        print " ", "M. Jones     ", mj
23        print " ", "Cust Fdbk Ofc", cfo
24        print " ", "==============", "======"
25        print " ", "TOTAL      ", total
26  }
```

```
 1  BEGIN {
 2        OFS = "\t"
 3        lmb=gsr=sjs=jc=bj=rt=ea=mj=cfo=0
 4  }
 5  $1 == "LMB" {lmb++}
 6  $1 == "GSR" {gsr++}
 7  $1 == "SJS" {sjs++}
 8  $1 == "JC"  {jc++}
 9  $1 == "BJ"  {bj++}
10  $1 == "RT"  {rt++}
11  $1 == "EA"  {ea++}
12  $1 == "MJ"  {mj++}
13  $1 == "CFO" {cfo++}

14  END { total=lmb+gsr+sjs+jc+bj+rt+ea+mj+cfo

15        print lmb
16        print gsr
17        print sjs
18        print jc
19        print bj
20        print rt
21        print ea
22        print mj
23        print cfo
24        print "======"
25        print total
26  }
```

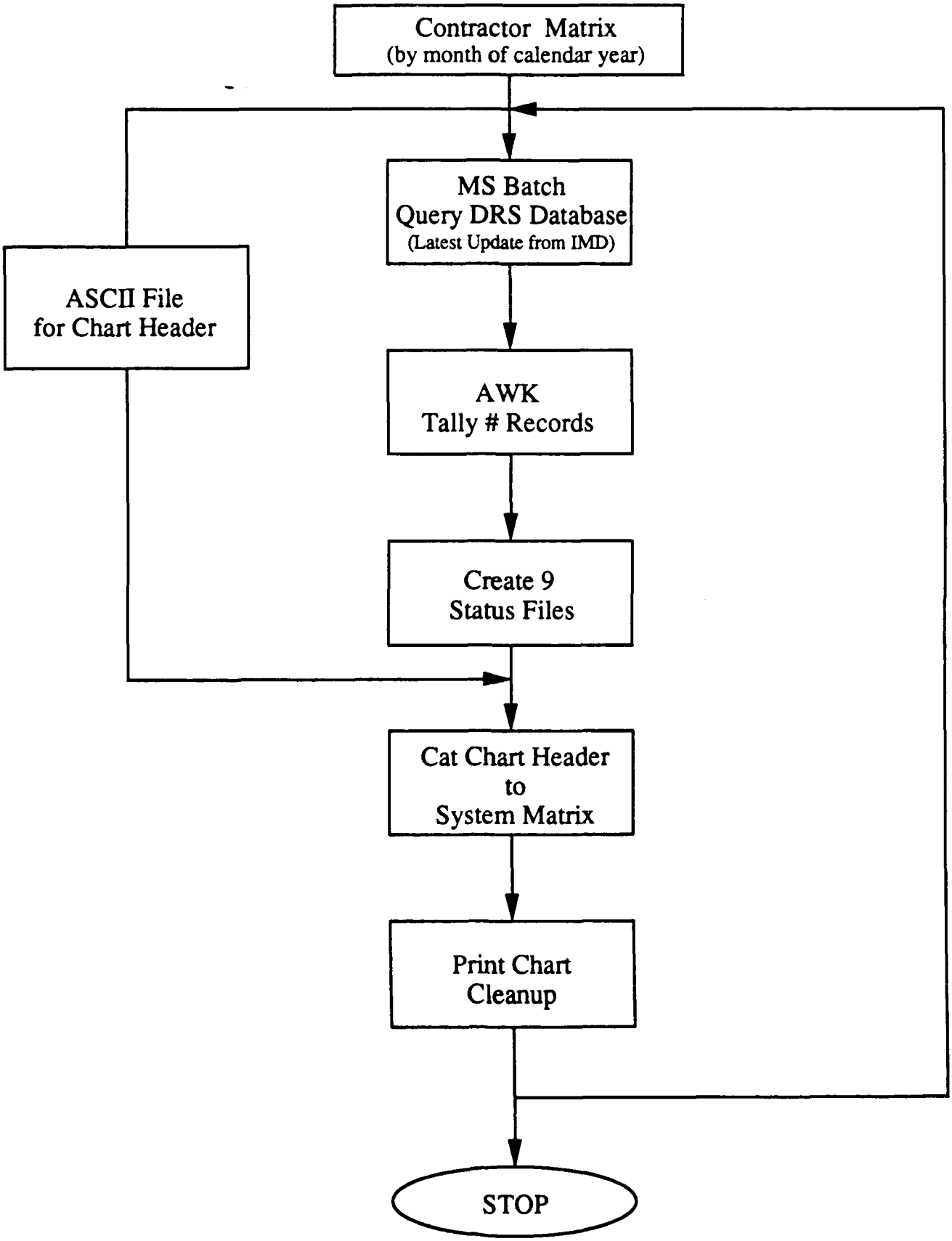
```
 1  dta=`date | cut -c5-7`
 2  dtb=`date | cut -c25-28`
 3  dtc=`date | cut -c9-10`
 4  #sumpd=`cat $1`
 5  cut -c1-2 $1 >zmo2
 6  cut -c7-10 $1 >zyr2
 7  zmo=`cat zmo2`
 8  zyr=`cat zyr2`
 9  if test $zmo = 01
10  then sumpd=JANUARY
11       sp=__
12  fi
13  if test $zmo = 02
14  then sumpd=FEBRUARY
15       sp=_
16  fi
17  if test $zmo = 03
18  then sumpd=MARCH
19       sp=____
20  fi
21  if test $zmo = 04
22  then sumpd=APRIL
23       sp=____
24  fi
25  if test $zmo = 05
26  then sumpd=MAY
27       sp=______
28  fi
29  if test $zmo = 06
30  then sumpd=JUNE
31       sp=_____
32  fi
33  if test $zmo = 07
34  then sumpd=JULY
35       sp=_____
36  fi
37  if test $zmo = 08
38  then sumpd=AUGUST
39       sp=___
40  fi
41  if test $zmo = 09
42  then sumpd=SEPTEMBER
43  fi
44  if test $zmo = 10
45  then sumpd=OCTOBER
46       sp=__
47  fi
48  if test $zmo = 11
49  then sumpd=NOVEMBER
50       sp=_
51  fi
52  if test $zmo = 12
53  then sumpd=DECEMBER
54       sp=_
55  fi
56  rm zmo2 zyr2

57  echo;
58  echo "        QUALITY DEFICIENCY REPORT SUMMARY"
59  echo "                    AMSMI-QA-CF                    SUMMARY PERIOD :
```

```
60  echo "
61  echo;
62  echo;
63  echo "        ACTION          OBOP    RCVD    CLOSED  TRANSF  OEOP    1-
64  echo "        OFFICER                                                 DA
65  echo "        ==============  ======  ======  ======  ======  ======  ==
```

APPENDIX F

Contractor QDR Summary Source Code

CONTRACTOR QDR SUMMARY



```
msbatch /usr2/acct/tnewby/drs/ndrs << }
set MSVALSEP to '       ';

select s2krg0.fscm_dfp, s2krg800.cntr, docu_mgt_no from s2krg0, s2krg800

select s2krg0.fscm_dfp, s2krg800.cntr, docu_mgt_no from s2krg0, s2krg800

select s2krg0.fscm_dfp, s2krg800.cntr, docu_mgt_no from s2krg0, s2krg800

select s2krg0.fscm_dfp, s2krg800.cntr, docu_mgt_no from s2krg0, s2krg800

select s2krg0.fscm_dfp, s2krg800.cntr, docu_mgt_no from s2krg0, s2krg800

select s2krg0.fscm_dfp, s2krg800.cntr, docu_mgt_no from s2krg0, s2krg800

select s2krg0.fscm_dfp, s2krg800.cntr, docu_mgt_no from s2krg0, s2krg800

select s2krg0.fscm_dfp, s2krg800.cntr, docu_mgt_no from s2krg0, s2krg800

select s2krg0.fscm_dfp, s2krg800.cntr, docu_mgt_no from s2krg0, s2krg800
}

dt=$2
echo $dt >zdt2
cut -c1-4 zdt2 >zyr
cut -c5-6 zdt2 >zmo
cut -c7-8 zdt2 >zda
slsh="/"
echo $slsh >zslsh
paste zmo zslsh >z2mo
paste zda zslsh >z2da
paste z2mo z2da zyr >zdate
sed 's, ,,g' zdate >z2date

sort OBOP >OBOPs
sort RCDP >RCDPs
sort CDRP >CDRPs
sort TDRP >TDRPs
sort OEOP >OEOPs
sort O1_120 >O1_120s
sort O120_180 >O120_180s
sort OV180 >OV180s
sort OV180C >OV180Cs

awk -f 0912 OBOPs >cntr_OBOP
heading3 z2date OBOP >zheading
cat zheading cntr_OBOP >31.$3

awk -f 0912 RCDPs >cntr_RCDP
heading3 z2date RCDP >zheading
cat zheading cntr_RCDP >32.$3

awk -f 0912 CDRPs >cntr_CDRP
heading3 z2date CDRP >zheading
cat zheading cntr_CDRP >33.$3

awk -f 0912 TDRPs >cntr_TDRP
heading3 z2date TDRP >zheading
cat zheading cntr_TDRP >34.$3
```

```
45  awk -f 0912 OEOPs >cntr_OEOP
46  heading3 z2date OEOP >zheading
47  cat zheading cntr_OEOP >35.$3

48  awk -f 0912 O1_120s >cntr_O1_120
49  heading3 z2date O1_120 >zheading
50  cat zheading cntr_O1_120 >36.$3

51  awk -f 0912 O120_180s >cntr_O120_180
52  heading3 z2date O120_180 >zheading
53  cat zheading cntr_O120_180 >37.$3

54  awk -f 0912 OV180s >cntr_OV180
55  heading3 z2date OV180 >zheading
56  cat zheading cntr_OV180 >38.$3

57  awk -f 0912 OV180Cs >cntr_OV180C
58  heading3 z2date OV180C >zheading
59  cat zheading cntr_OV180C >39.$3

60  rm zdt2 zmo zda zyr z2mo z2da zslsh zdate z2date zheading
61  rm CDRP O120_180 O1_120 OBOP OEOP OV180 OV180C RCDP TDRP
62  rm CDRPs O120_180s O1_120s OBOPs OEOPs OV180s OV180Cs RCDPs TDRPs
63  rm cntr_CDRP cntr_O120_180 cntr_O1_120 cntr_OBOP cntr_OEOP cntr_OV180
64  rm cntr_OV180C cntr_RCDP cntr_TDRP
```

```
 1  BEGIN {FS = "\t"}
 2  {
 3  #{print "$1 : "$1"  ""$2 : "$2"  ""prev : "prev"  ""cc : "cc}
 4  if ($1 != prev)
 5  {
 6    printf("\t%s\t%s\t\t%s\n", prev, cntr, cc)
 7    cc= 1
 8    prev= $1
 9    cntr= $2
10  }
11  else
12    cc++
13  }
14  END {
15  printf("\t%s\t%s\t\t%s\n", prev, cntr, cc)}
```

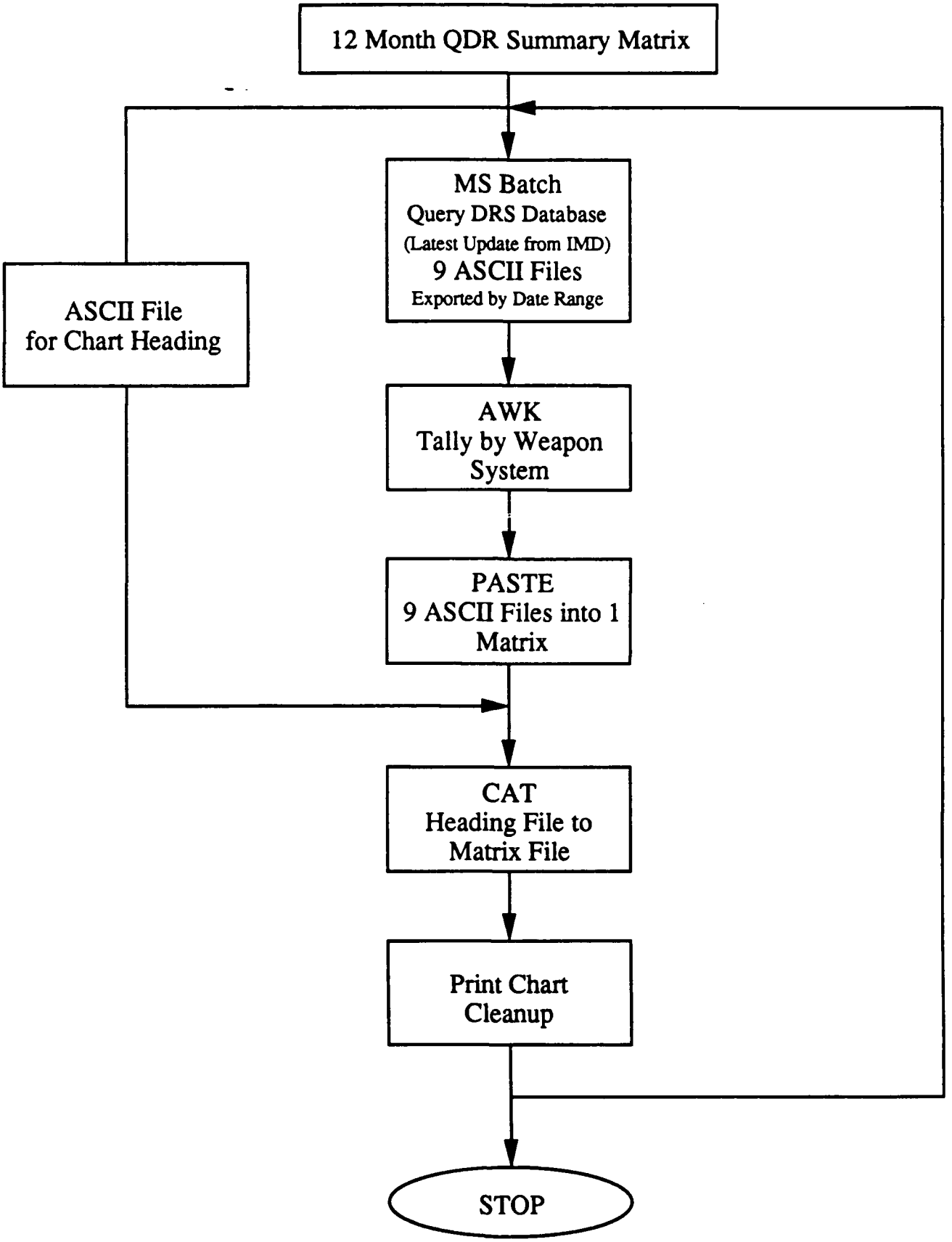
```
 1  dta=`date | cut -c5-7`
 2  dtb=`date | cut -c25-28`
 3  dtc=`date | cut -c9-10`
 4  #sumpd=`cat $1`
 5  cut -c1-2 $1 >zmo2
 6  cut -c7-10 $1 >zyr2
 7  zmo=`cat zmo2`
 8  zyr=`cat zyr2`

 9  if test $zmo = 01
10  then sumpd=JANUARY
11       sp=__
12  fi
13  if test $zmo = 02
14  then sumpd=FEBRUARY
15       sp=_
16  fi
17  if test $zmo = 03
18  then sumpd=MARCH
19       sp=____
20  fi
21  if test $zmo = 04
22  then sumpd=APRIL
23       sp=____
24  fi
25  if test $zmo = 05
26  then sumpd=MAY
27       sp=___ __
28  fi
29  if test $zmo = 06
30  then sumpd=JUNE
31       sp=_____
32  fi
33  if test $zmo = 07
34  then sumpd=JULY
35       sp=_____
36  fi
37  if test $zmo = 08
38  then sumpd=AUGUST
39       sp=___
40  fi
41  if test $zmo = 09
42  then sumpd=SEPTEMBER
43  fi
44  if test $zmo = 10
45  then sumpd=OCTOBER
46       sp=__
47  fi
48  if test $zmo = 11
49  then sumpd=NOVEMBER
50       sp=_
51  fi
52  if test $zmo = 12
53  then sumpd=DECEMBER
54       sp=_
55  fi
56  rm zmo2 zyr2

57  if test $2 = OBOP
58  then title="OBOP"
```

```
59  fi
60  if test $2 = RCDP
61  then title="RCVD"
62  fi
63  if test $2 = CDRP
64  then title="CLOSED"
65  fi
66  if test $2 = TDRP
67  then title="TRANSF"
68  fi
69  if test $2 = OEOP
70  then title="OEOP"
71  fi
72  if test $2 = O1_120
73  then title="1-119 DAYS"
74  fi
75  if test $2 = O120_180
76  then title="120-180 DAYS"
77  fi
78  if test $2 = OV180
79  then title="OVER 180"
80  fi
81  if test $2 = OV180C
82  then title="OVR 180 CLOSED"
83  fi
84  echo;
85  echo "\tCONTRACTOR - QUALITY DEFICIENCY REPORT SUMMARY"
86  echo "\t\t\t\t AMSMI-QA-CF"
87  echo
88  echo "\t\t\t\t      SUMMARY PERIOD : ""$sumpd"" ""$zyr"" ""$sp";
89  echo "\t\t\t\t                       (as of $dtc" "$dta" "$dtb)";
90  echo
91  echo
92  echo "\tCNTR"
93  echo "\tCODE\t\t    CONTRACTOR\t\t\t$title"
94  echo "\t=====\t=================================\t=============="
```

APPENDIX G

QDR Status Report Source Code

QDR STATUS REPORT



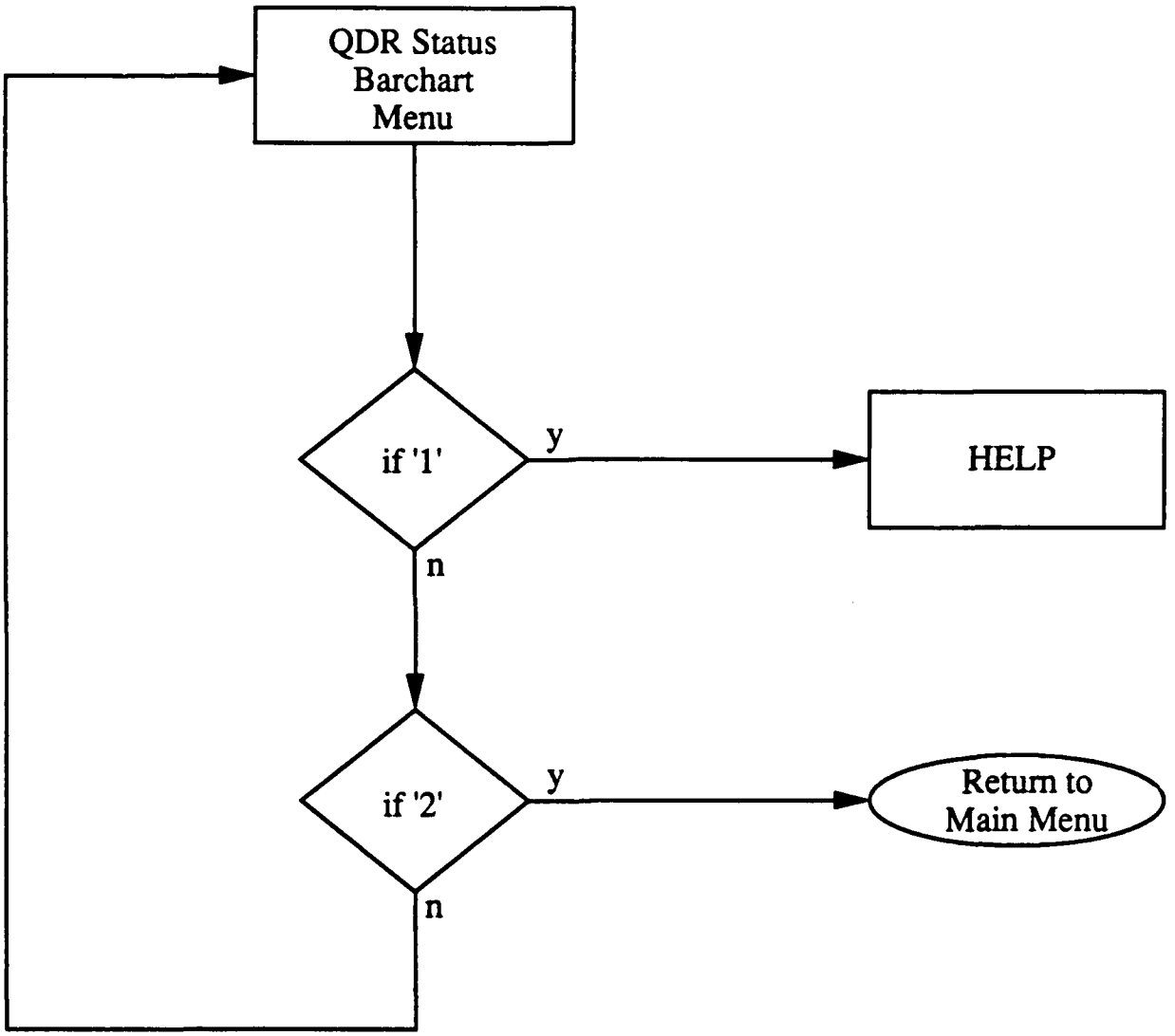
```
msbatch /usr2/acct/tnewby/drs/ndrs << }
set MSVALSEP to '       ';

select wpn_sys_cd from s2krg0 dump into OBOP where docu_mgt_no match "P*

select wpn_sys_cd from s2krg0 dump into RCDP where docu_mgt_no match "P*

select wpn_sys_cd from s2krg0 dump into REOP where docu_mgt_no match "P*

select wpn_sys_cd from s2krg0 dump into CDRP where docu_mgt_no match "P*

select wpn_sys_cd from s2krg0 dump into TDRP where docu_mgt_no match "P*

select wpn_sys_cd from s2krg0 dump into OEOP where docu_mgt_no match "P*

select wpn_sys_cd from s2krg0 dump into O1_120 where docu_mgt_no match "

select wpn_sys_cd from s2krg0 dump into O120_180 where docu_mgt_no match

select wpn_sys_cd from s2krg0 dump into OV180 where docu_mgt_no match "P

select wpn_sys_cd from s2krg0 dump into OV180C where docu_mgt_no match "

}
awk -f 0509 OBOP > ALLSYS
awk -f 0509 RCDP >> ALLSYS
awk -f 0509 REOP >> ALLSYS
awk -f 0509 CDRP >> ALLSYS
awk -f 0509 TDRP >> ALLSYS
awk -f 0509 OEOP >> ALLSYS
awk -f 0509 O1_120 >> ALLSYS
awk -f 0509 O120_180 >> ALLSYS
awk -f 0509 OV180 >> ALLSYS
awk -f 0509 OV180C >> ALLSYS

cp ALLSYS $3$4.num
cat $3.t $4.t > YR.a
cat YR.a LINE > YR.b
cat YR.b $3$4.num > $3$4.CFO

ls -t *.CFO > date.in
split -1 date.in
cat xal xak xaj xai xah xag > date.ina
cat xaf xae xad xac xab xaa > date.inb
hor_lsta=`paste -s -d" " date.ina`
hor_lstb=`paste -s -d" " date.inb`
paste AL $hor_lsta >half.cur
paste half.cur $hor_lstb >$3$4.sum

TOP >ZTOP

cat ZTOP $3$4.sum > $5

rm ALLSYS OBOP RCDP REOP CDRP TDRP OEOP O1_120 O120_180 OV180 OV180C
rm $3$4.num $3$4.sum half.cur YR.* xa* date.in* ZTOP
```

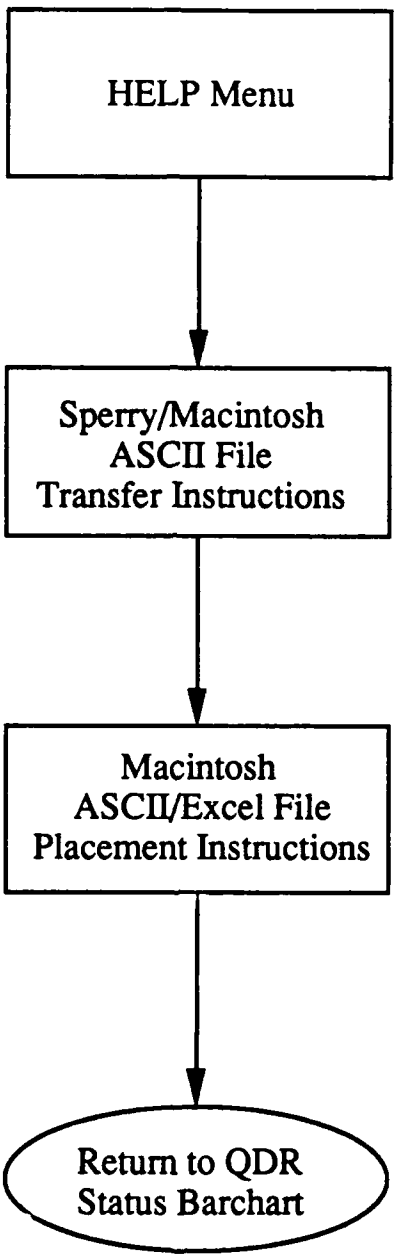
```
1  END {
2       print " "
3       print NR
4       print "_____"
5   }
```

APPENDIX H

QDR Status Barchart Menu Source Code

FILE DOWNLOADING INSTRUCTIONS (xcode_c)




HELP Menu
Sperry/Macintosh
ASCII File
Transfer Instructions
Macintosh
ASCII/Excel File
Placement Instructions
Return to QDR
Status Barchart

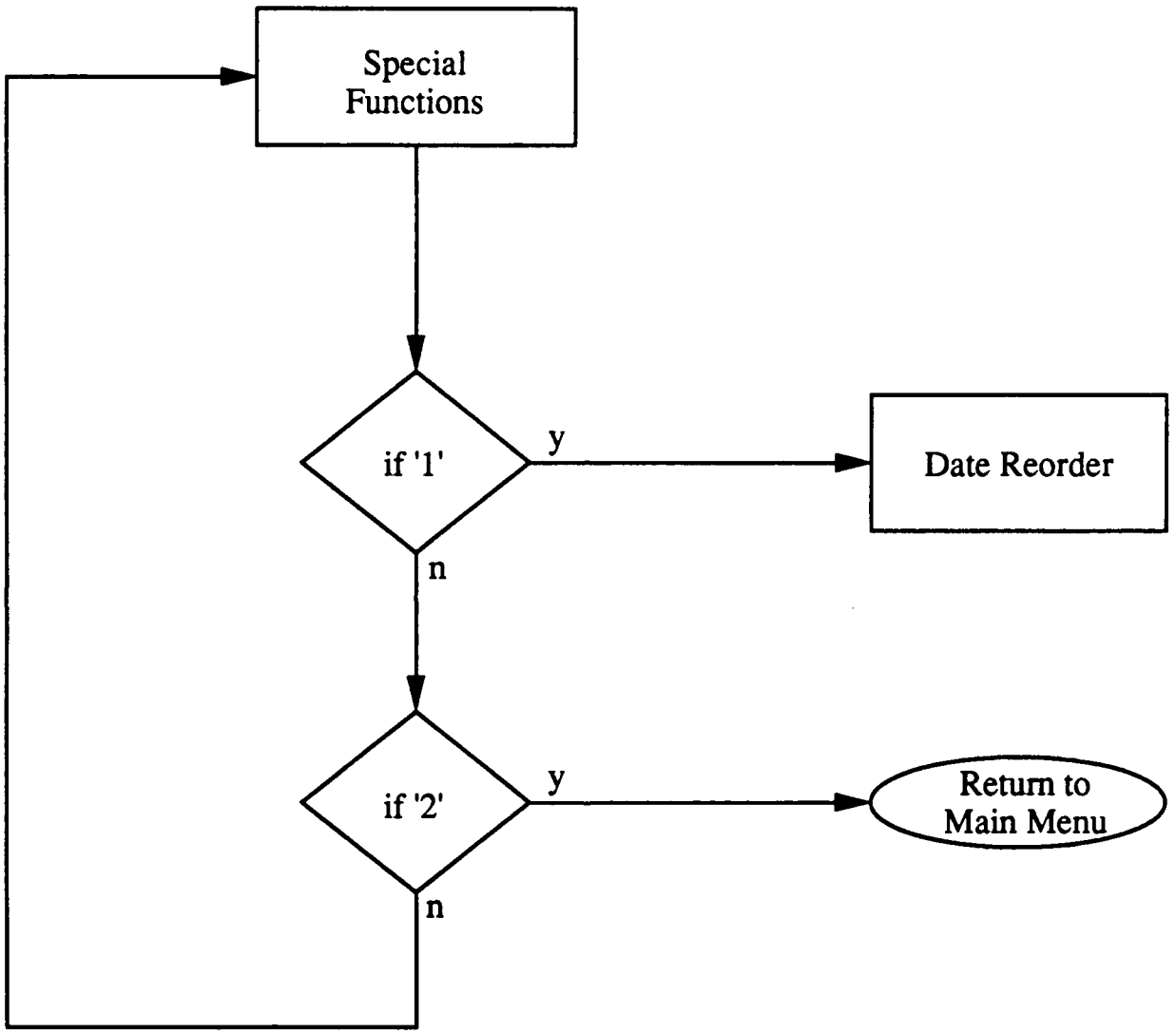
```
 1  sel=
 2  while (:)
 3  clear
 4  echo;
 5  echo "\t\t\t    QDR STATUS BARCHART MENU"
 6  echo "\t_____________________________________________________________"
 7  echo;
 8  echo;
 9  echo "\t\t\t\t1-  Help"
10  echo "\t\t\t\t2-  Main Menu"
11  echo;
12  echo;
13  echo "\tEnter Choice : \c"
14  read choice

15  do
16  case "$choice" in
17  1)
18  clear;
19  echo;
20  echo "\t       Sperry/Macintosh ASCII File Transfer Instructions"
21  echo "\t_____________________________________________________________"
22  echo
23  echo
24  echo "\t 1.  At the UNIX prompt,              Type : kermit"
25  echo
26  echo "\t 2.  When the prompt 'C-Kermit>'"
27  echo "\t     is present,                      Type : send filename"
28  echo "\t                                      enter"
29  echo
30  echo "\t 3.  To set up Red Ryder,            move pointer to the top"
31  echo "\t                                      status bar, drag on FILE, an
32  echo
33  echo "\t                                      Select : receive file - Kerm
34  echo "\t                                      Enter : receive filename"
35  echo
36  echo "\t 4.  When the downloading"
37  echo "\t     procedure is completed,          move pointer to the top"
38  echo "\t                                      status bar, drag on FILE, an
39  echo
40  echo "\t                                      Select : quit\c"
41  read pause
42  clear;
43  echo
44  echo "\t       Macintosh ASCII/Excel File Placement Instructions"
45  echo "\t_____________________________________________________________"
46  echo
47  echo "\t 1.  Locate the folder containing the QDR related files."
48  echo
49  echo "\t 2.  Move pointer to the 'QDR_MASTER_XCL' excel file and click."
50  echo "\t      (Note: NEVER alter or lose the 'QDR_MASTER_XCL' file!)"
51  echo
52  echo "\t 3.  Move pointer to the MENU BAR, click on FILE, and choose"
53  echo "\t      DUPLICATE."
54  echo
55  echo "\t 4.  Move pointer to the 'copy of QDR_MASTER_XCL' excel file and
56  echo "\t     rename it (e.g., '3Q87_3Q89')."
57  echo
58  echo "\t 5.  Move pointer to the new renamed excel file and select"
59  echo "\t      (i.e., double-click) to open."
```

```
60  echo
61  echo "\t 6.  Once opened and inside the spreadsheet, move pointer to the
62  echo "\t     fiscal year labels and update if necessary."
63  echo
64  echo "\t 7.  Move pointer to cell B1, click, and while holding down the"
65  echo "\t     mouse button, drag the pointer to cell AB243 (243R x 27C).
66  read pause
67  clear
68  echo
69  echo "\t    Macintosh ASCII/Excel File Placement Instructions (Con't)"
70  echo "\t_____________________________________________________________________
71  echo
72  echo "\t 8.  Move pointer to MENU BAR, click on FILE, choose OPEN and"
73  echo "\t      select the ASCII downloaded file (e.g., 3Q87_3Q89.mac)."
74  echo
75  echo "\t 9.  Once opened, move pointer to cell A1, click, and while hold
76  echo "\t      down the mouse button, drag the pointer to cell AA243."
77  echo
78  echo "\t10.  Move pointer to the MENU BAR, click on EDIT, and choose COP
79  echo
80  echo "\t11.  Move pointer to box at top left-hand corner and click to cl
81  echo "\t      the file. Respond to the displayed message by choosing"
82  echo "\t      'Save Formatted Values'."
83  echo
84  echo "\t12.  Move pointer to MENU BAR, click on EDIT, and choose PASTE."
85  echo "\t      (Note: the ASCII file will now be loaded into Excel"
86  echo
87  echo "\t13.  When successfully loaded, move pointer to box at top left-h
88  echo "\t      corner, click to close the file, and choose the SAVE optio
89  echo
90  echo "\t14.  This completes the required steps for preparing the weapon"
91  echo "\t      system data to create the Macintosh generated barcharts.\c
92  read pause
93  ;;

94  2)
95  clear;
96  break;
97  ;;

98  esac
99  done
```

APPENDIX I

Special Functions Menu Source Code

SPECIAL FUNCTIONS (xcode_d)

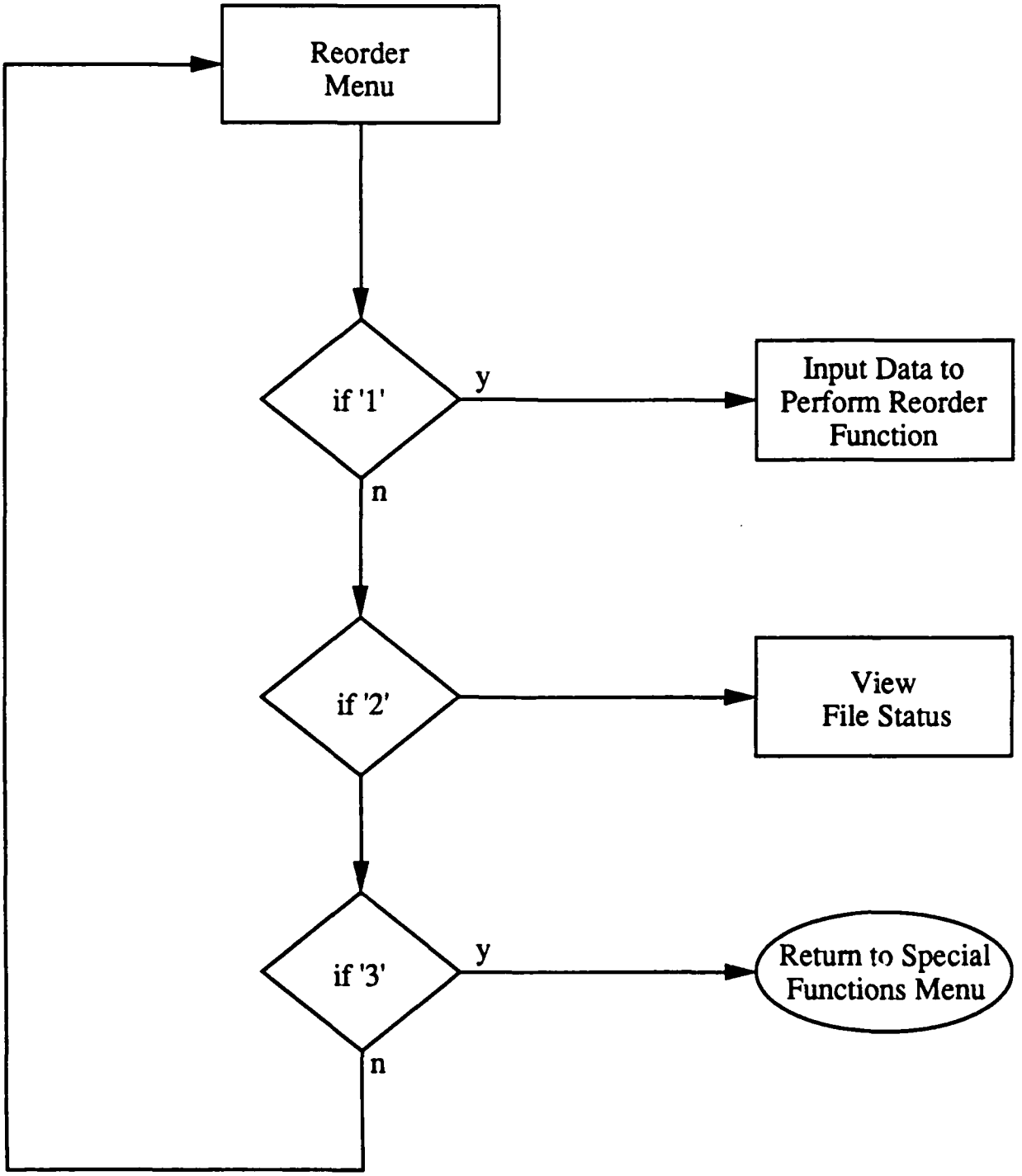


```
 1  sel=
 2  while (:)
 3  clear
 4  echo;
 5  echo "\t\t\t     SPECIAL FUNCTIONS MENU"
 6  echo "\t_____________________________________________________________
 7  echo;
 8  echo;
 9  echo "\t\t\t\t1-  Date Reorder"
10  echo "\t\t\t\t2-  Main Menu"
11  echo;
12  echo;
13  echo "\tEnter Choice : \c"
14  read choice

15  do
16  case "$choice" in
17  1)
18  clear;
19  xcode_a;
20  clear;
21  ;;
22  2)
23  clear;
24  break;
25  ;;

26  esac
27  done
```

APPENDIX J

Reorder Menu Source Code

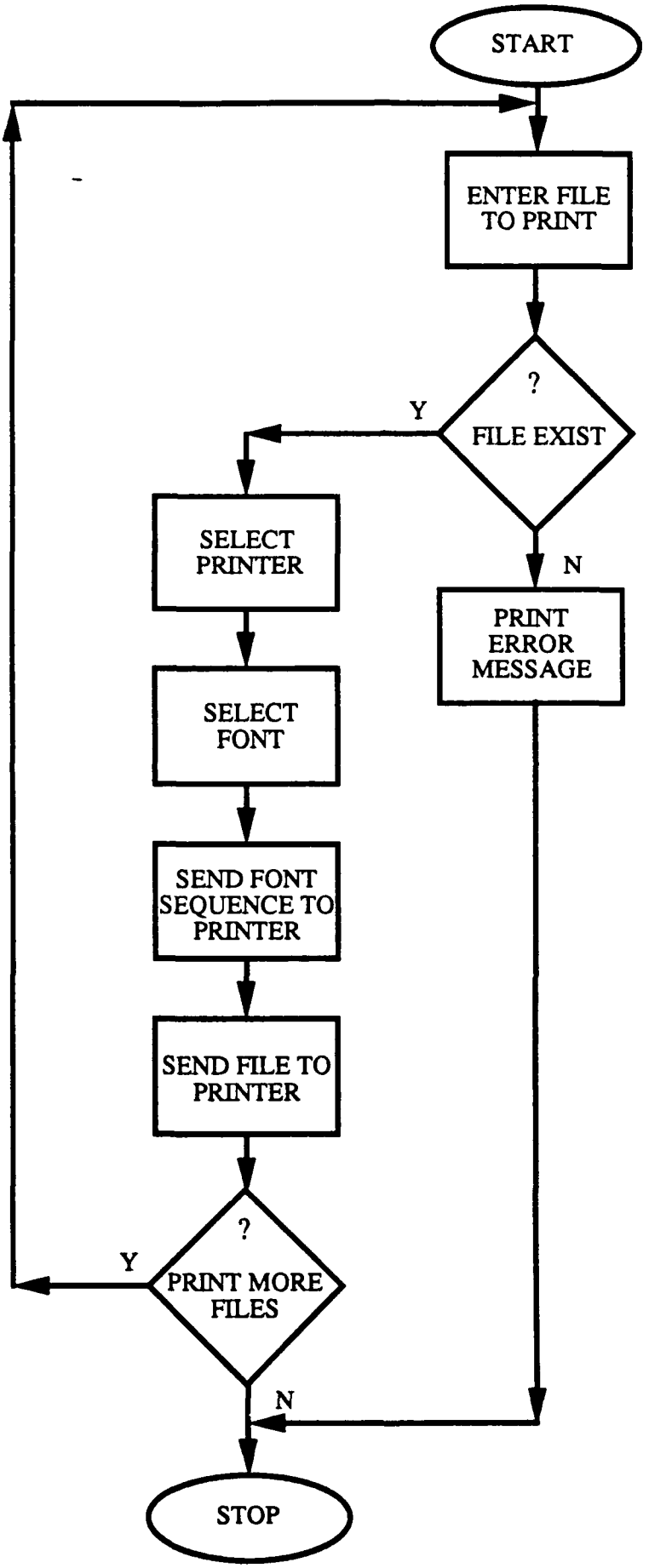
DATE REORDER



APPENDIX M

Laser Printer/Font Program

Laser Printer/Font Menu



```
##                          LASER PRINTER / FONT PROGRAM
##
prt=0
clear
##
## BEGIN MOREFILE LOOP
##
while [ "$prt" = 0 ]
do
echo;
echo;
echo "\t\tLASER PRINTER / FONT PROGRAM"
echo "\t\t--------------------------"
echo;
echo "\t\tEnter filename : \c"
read fname
##
## CHECK IF GIVEN FILENAME EXISTS
##
if test ! -s $fname
then
        echo;
        echo "\t\t **** No file < $fname > ****"
        exit
fi
##
## END FILENAME CHECK
##
i=0
##
## BEGIN PRINTER CHOICE LOOP
##
while [ "$i" = 0 ]
do
clear
echo;
echo "     File to print : $fname"
echo "                   ----------"
echo;
echo "\t*************************************************************************"
echo "\t*                                                                       *"
echo "\t*                    SPERRY 5000 - LASER PRINTERS                       *"
echo "\t*                  ---------------------------------                    *"
echo "\t*                                                                       *"
echo "\t*                              1) CFLASER                               *"
echo "\t*                              2) CCLASER                               *"
echo "\t*                              3) QDLASER                               *"
echo "\t*                                                                       *"
echo "\t*************************************************************************"
echo;
echo "\t\tEnter number of laser to print : \c"
read lnum
```

```
case $lnum in

1)
las=cflaser
lasr=cfolaser
i=1
;;

2)
las=cclaser
lasr=ccolaser
i=1
;;

3)
las=qdlaser
lasr=qdolaser
i=1
;;

*)
echo;
echo "              **** INVALID RESPONSE ****"
SLEEP 2
;;

esac
done
##
## END PRINTER CHOICE LOOP
##
clear
j=0
##
## BEGIN FONT CHOICE LOOP
##
while [ "$j" = 0 ]
do
clear
dta='date | cut -c-10'
dtb='date | cut -c25-28'
tt='date | cut -c12-16'
echo;
echo "\t$dta", "$dtb"
echo "\t$tt"
```

```
echo "\t*****************************************************************"
echo "\t*                                                               *"
echo "\t*                     LASER PRINTER                             *"
echo "\t*                       FONT MENU                               *"
echo "\t*   ----------------------------------------------------------- *"
echo "\t*     FONT                   ORINT     POINT      PITCH         *"
echo "\t*   - --------------         ----------  ----------  --------- *"
echo "\t*   1) COURIER                 P         12          10         *"
echo "\t*   2) PRESTIGE ELITE          P         10          12         *"
echo "\t*   3) PRESTIGE ELITE          P         7.2         16.6       *"
echo "\t*   4) LINE PRINTER            P         9           16.6       *"
echo "\t*   5) COURIER                 L         12          10         *"
echo "\t*   6) PRESTIGE ELITE          L         10          12         *"
echo "\t*   7) PRESTIGE ELITE          L         7.2         16.6       *"
echo "\t*   8) LINE PRINTER            L         9           16.6       *"
echo "\t*****************************************************************"
echo;
echo "\t\tEnter Font Choice : \c"
read fnum

case $fnum in

1)
fnt=1
j=1
;;

2)
fnt=6
j=1
;;

3)
fnt=7
j=1
;;

4)
fnt=15
j=1
;;

5)
fnt=17
j=1
;;

6)
fnt=23
j=1
;;
```

```
7)
fnt=25
j=1
;;

8)
fnt=34
j=1
;;

*)
echo;
echo "              INVALID FONT CHOICE -- Please Reenter"
;;

esac
done
##
## END FONT CHOICE LOOP
##

clear
echo;
echo;
echo "\t\t\t**** PRINTING IN PROGRESS ****"
echo;
cat font$fnt | lp -d$lasr
##
## SEND FONT CHANGE SEQUENCE TO PRINTER
##
sleep 24
cat $fname | lp -d$las
##
## SEND FILE TO PRINTER
##
```

```
echo;
echo;
echo "\t\tPrint another file (Y/N) : \c"
read $morefile in

[Yy])
clear
## RESTART PROGRAM FROM BEGINNING
;;

[Nn])
clear
prt=1
## EXIT PROGRAM
;;

esac
done
##
## END MOREFILE LOOP
exit
## EXIT PROGRAM
```

```
 1  sel=
 2  while (:)
 3  clear
 4  echo;
 5  echo "\t\t\t\tREORDER MENU"
 6  echo "\t_____________________________________________________________
 7  echo;
 8  echo;
 9  echo "\t\t\t\t1-  Input Data"
10  echo "\t\t\t\t2-  File Status"
11  echo "\t\t\t\t3-  Main Menu"
12  echo;
13  echo;
14  echo "\tEnter Choice : \c"
15  read choice

16  do
17  case "$choice" in
18  1)
19  clear;
20  echo "\t\tThis utility allows the user to split, reorder, and merge"
21  echo "\t a specified field in a database table."
22  echo;
23  echo "\t\tShell Script Parameters : "
24  echo;
25  echo "\t\tp1- table name with fields separated by ^V."
26  echo "\t\tp2- field position in table."
27  echo "\t\tp3- first cut starting position in field."
28  echo "\t\tp4- first cut character length in field."
29  echo "\t\tp5- second cut starting position in field."
30  echo "\t\tp6- second cut character length in field."
31  echo "\t\tp7- table name of reordered field."
32  echo;
33  echo "\t\tEnter p1        : \c"
34  read p1;
35  echo $p1
36  echo "\t\tEnter p2        : \c"
37  read p2;
38  echo $p2
39  echo "\t\tEnter p3 and p4 : \c"
40  read p3 p4
41  echo $p3 $p4
42  echo "\t\tEnter p5 and p6 : \c"
43  read p5 p6;
44  echo $p5 $p6
45  echo "\t\tEnter p7        : \c"
46  read p7;
47  echo $p7
48  sleep 8;
49  clear;
50  echo;
51  echo "\t\tInformation is being processed, please wait ..."
52  echo;
53  echo;
54  xdr $p1 $p2 $p3 $p4 $p5 $p6 $p7
55  clear;
56  ;;
57  3)
58  clear
59  break
```

```
60  ;;
61  2)
62  clear;
63  echo;
64  echo;
65  l $p7
66  sleep 8;
67  ;;

68  esac
69  done
```

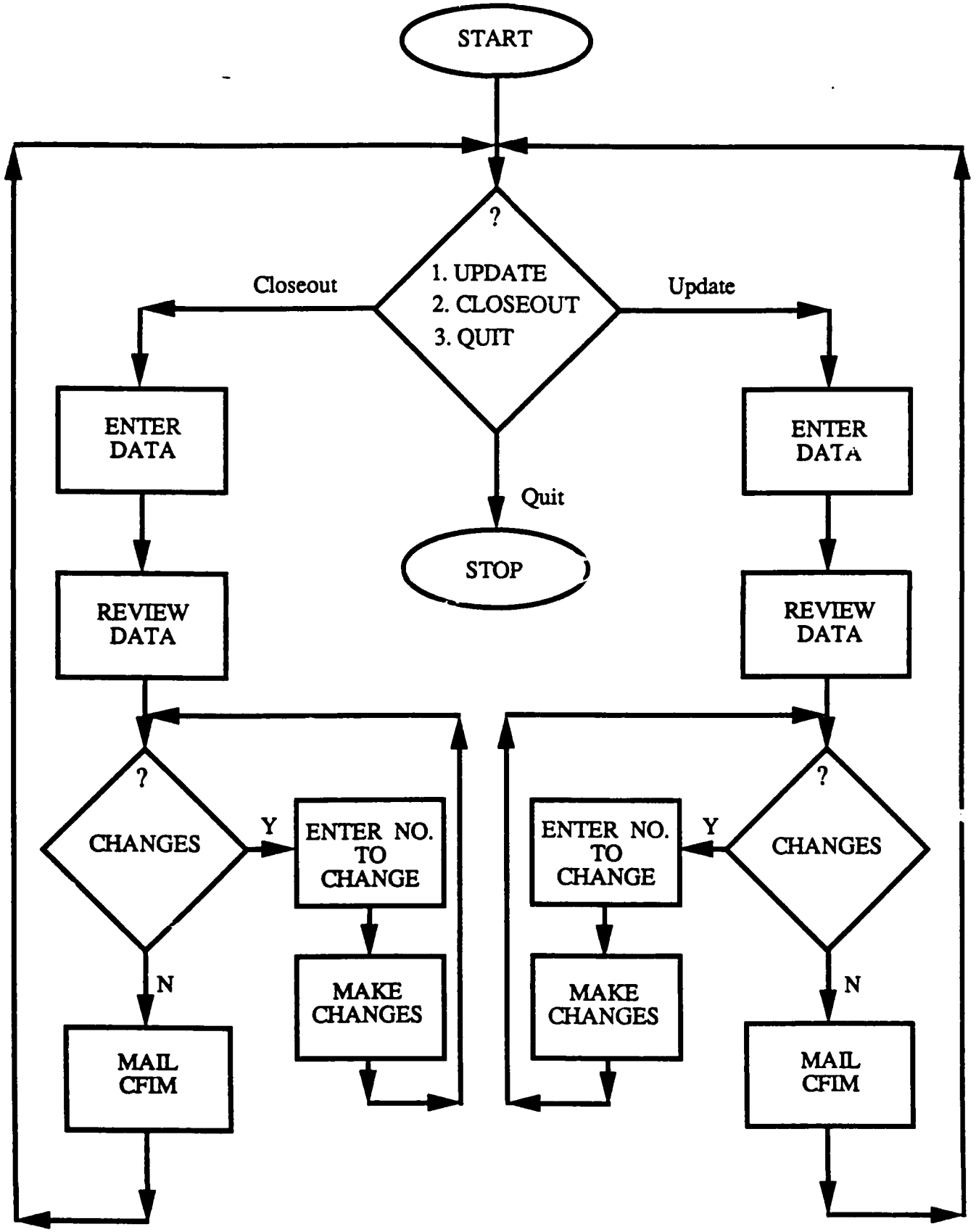
```
BEGIN {
      OFS = "\t"
      bx=dc=a2c=de=ds=dh=dj=dr=d6=d7=ef=en=ep=ey=e2=e3=e5=e7=0
      fa=fc=fg=fp=fq=fr=f3=f4=mw=h4=oo=td=xx=0
      a22=a23=a25=a2f=a2g=a2h=a2n=a2p=a2r=a2s=0
}
$1 == "BX" {bx++}
$1 == "DC" {dc++}
$1 == "2C" {a2c++}
$1 == "DE" {de++}
$1 == "DS" {ds++}
$1 == "DH" {dh++}
$1 == "DJ" {dj++}
$1 == "DR" {dr++}
$1 == "D6" {d6++}
$1 == "D7" {d7++}
$1 == "EF" {ef++}
$1 == "EN" {en++}
$1 == "EP" {ep++}
$1 == "EY" {ey++}
$1 == "E2" {e2++}
$1 == "E3" {e3++}
$1 == "E5" {e5++}
$1 == "E7" {e7++}
$1 == "FA" {fa++}
$1 == "FC" {fc++}
$1 == "FG" {fg++}
$1 == "FP" {fp++}
$1 == "FQ" {fq++}
$1 == "FR" {fr++}
$1 == "F3" {f3++}
$1 == "F4" {f4++}
$1 == "MW" {mw++}
$1 == "XX" {xx++}
$1 == "HD" {hd++}
$1 == "TD" {td++}
$1 == "00" {oo++}
$1 == "22" {a22++}
$1 == "23" {a23++}
$1 == "25" {a25++}
$1 == "2F" {a2f++}
$1 == "2G" {a2g++}
$1 == "2H" {a2h++}
$1 == "2N" {a2n++}
$1 == "2P" {a2p++}
$1 == "2R" {a2r++}
$1 == "2S" {a2s++}
END { tot=bx+dc+a2c+de+ds+a2s+dh+a2h+dj+dr+a2r+d6+d7+ef+a2f+en+a2n+ep+a2
      tot1=ey+e2+a22+e3+a23+e5+a25+e7+fa+fc+fg+a2g+fp+fq+fr+f3+f4+mw
      tot2=td+h4+oo+xx
      total=tot+tot1+tot2
      print bx
      print (dc + a2c)
      print (de + ds + a2s)
      print (dh + a2h)
      print dj
      print (dr + a2r)
      print d6
      print d7
      print (ef + a2f)
```

```
61          print (en + a2n)
62          print (ep + a2p)
63          print ey
64          print (e2 + a22)
65          print (e3 + a23)
66          print (e5 + a25)
67          print e7
68          print fa
69          print fc
70          print (fg + a2g)
71          print fp
72          print fq
73          print fr
74          print f3
75          print f4
76          print mw
77          print (td + h4 + oo + xx)
78          print total
79  }
```

APPENDIX K

Memo Utility for DRS Update

DRS UPDATE MENU



```
##              MAIN MENU PROGRAM FOR DRS UPDATE
##
## EXECUTE PROGRAM UNTIL <QUIT> IS CHOSEN
##
r=0
while [ "$r" = 0 ]
do
clear
echo "\033[5;33HDRS UPDATE MENU"
echo "\033[6;33H----------"
echo "\033[9;35H1. REMARK"
echo "\033[11;35H2. CLOSEOUT"
echo "\033[13;35H3. QUIT"
i=0
##
## BEGIN CHOOSE ACTION LOOP
##
while [ "$i" = 0 ]
do
echo "\033[16;30HENTER UPDATE ACTION : \c"
read choice
case $choice in

1)
update.txt
i=1
;;

2)
cl.txt
i=1
;;

3)
clear
i=1
r=1
exit
;;

*)
echo;
;;

esac
done
##
## END CHOOSE ACTION LOOP
##
done
## END PROGRAM LOOP
exit
```

```
## THIS ROUTINE SENDS MEMO'S TO CFO FOR CLOSEOUT ACTIONS
##
## The user is prompted for entries, then asked to verify the
## entries and then mail the memo to CFO.
##
##
## Initialize the variables
##
clear
dmn=" "
ao=" "
typeqdr=" "
findiscod=" "
apc=" "
actdate=" "
dodaac=" "
cardseq1=" "
cardseq2=" "
cardseq3=" "
cardseq4=" "
cardseq5=" "
##
## THE ENTRY SCREEN FOR THE CLOSEOUT DATA
##
echo "\033[2;33HDRS CLOSE-OUT ACTION"
echo "\033[3;33H----------"
echo "\033[4;1HDocument Management Number: \c"
echo "\033[4;45HAction Officer : \c"
echo "\033[6;1HTYPE QDR CODE : \c"
echo "\033[6;37HFINAL DISPOSITION CODE : \c"
echo "\033[8;1HAction Process Code : \c"
echo "\033[8;36HDate of Action (MMDDYY) : \c"
echo "\033[10;1HCARD 01 : \c"
echo "\033[11;1HCARD 02 : \c"
echo "\033[12;1HCARD 03 : \c"
echo "\033[13;1HCARD 04 : \c"
echo "\033[14;1HCARD 05 : \c"
echo "\033[15;1HCARD 06 : \c"
echo "\033[16;1HCARD 07 : \c"
echo "\033[17;1HCARD 08 : \c"
echo "\033[4;30H\c"
read dmn
echo "\033[4;62H\c"
read ao
echo "\033[6;17H\c"
read typeqdr
echo "\033[6;62H\c"
read findiscod
echo "\033[8;23H\c"
read apc
echo "\033[8;62H\c"
read actdate
```

```
echo "\033[10;11H\c"
read cardseq1
echo "\033[11;11H\c"
read cardseq2
echo "\033[12;11H\c"
read cardseq3
echo "\033[13;11H\c"
read cardseq4
echo "\033[14;11H\c"
read cardseq5
echo "\033[15;11H\c"
read cardseq6
echo "\033[16;11H\c"
read cardseq7
echo "\033[17;11H\c"
read cardseq8
##
## THE DATA SCREEN FOR VERIFICATION
##
j=0
##
## BEGINNING OF VERIFICATION LOOP
##
while [ "$j" = 0 ]
do
clear
echo "\033[1;32HDRS CLOSE-OUT MEMO"
echo "\033[2;32H------------------"
echo "\033[4;2H1) DOCUMENT MANAGEMENT NUMBER : $dmn"
echo "\033[4;45H2) ACTION OFFICER : $ao"
echo "\033[5;34H---------\033[5;68H----"
echo "\033[6;2H3) TYPE QDR CODE : $typeqdr"
echo "\033[6;37H4) FINAL DISPOSITION CODE : $findiscod"
echo "\033[7;21H----\033[7;65H----"
echo "\033[8;2H5) ACTION PROCESS CODE : $apc"
echo "\033[8;37H6) ACTION DATE : $actdate"
echo "\033[9;27H-\033[9;54H------"
echo "\033[11;1H 7) CARD 01 : $cardseq1"
echo "\033[12;1H 8) CARD 02 : $cardseq2"
echo "\033[13;1H 9) CARD 03 : $cardseq3"
echo "\033[14;1H10) CARD 04 : $cardseq4"
echo "\033[15;1H11) CARD 05 : $cardseq5"
echo "\033[16;1H12) CARD 06 : $cardseq6"
echo "\033[17;1H13) CARD 07 : $cardseq7"
echo "\033[18;1H14) CARD 08 : $cardseq8"
k=0
```

```
##
## BEGINNING OF MORE CHANGES LOOP
##
while [ "$k" = 0 ]
do
echo "\033[20;20HARE ENTRIES CORRECT (Y/N) ? \c"
read yn
case $yn in
[Yy])
##
## THE ROUTINE COPIES ALL DATA TO A FILE AND THEN MAILS THIS
## FILE TO THE CUSTOMER FEEDBACK OFFICE.
##
echo "DRS UPDATE ACTION " > > $dmn.$apc
echo "----------------- " > > $dmn.$apc
echo "DOCUMENT MANAGEMENT NUMBER : $dmn \t\tACTION OFFICER : $ao" > > $dmn.
echo " " > > $dmn.$apc
echo "TYPE QDR CODE : $typeqdr \t\t\tFINAL DISPOSITION CODE : $findiscod"
> > $dmn.$apc
echo " " > > $dmn.$apc
echo "ACTION PROCESS CODE : $apc \t\t\tDATE OF ACTION : $actdate" > > $dmn.$apc
echo " " > > $dmn.$apc
echo "CARD 01 : $cardseq1" > > $dmn.$apc
echo "CARD 02 : $cardseq2" > > $dmn.$apc
echo "CARD 03 : $cardseq3" > > $dmn.$apc
echo "CARD 04 : $cardseq4" > > $dmn.$apc
echo "CARD 05 : $cardseq5" > > $dmn.$apc
echo "CARD 06 : $cardseq2" > > $dmn.$apc
echo "CARD 07 : $cardseq3" > > $dmn.$apc
echo "CARD 08 : $cardseq4" > > $dmn.$apc
echo "= = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = =
mail cfim < $dmn.$apc
echo "\033[23;21H   *** SENDING MAIL ***"
sleep 2
k=1
j=1
;;

[Nn])
##
## THIS ROUTINE ASKS FOR THE NUMBER OF THE ENTRY TO CHANGE.
## LETS THE USER CHANGE THE ENTRY AND THEN REVERIFIES ENTRIES
## BEFORE MAILING
##
m=0
##
## BEGINNING OF ENTRY CHANGE LOOP
##
while [ "$m" = 0 ]
do
echo "\033[20;20HENTER NUMBER TO CHANGE : \c"
read num
```

```
case $num in

1)
echo "\033[4;34H\c"
read dmn
m=1
k=1
;;

2)
echo "\033[4;65H\c"
read ao
m=1
k=1
;;

3)
echo "\033[6;21H\c"
read typeqdr
m=1
k=1
;;

4)
echo "\033[6;65H\c"
read findiscod
m=1
k=1
;;

5)
echo "\033[8;27H\c"
read apc
m=1
k=1
;;

6)
echo "\033[8;54H\c"
read actdate
m=1
k=1
;;

7)
echo "\033[11;15H\c"
read cardseq1
m=1
k=1
;;
```

```
8)
echo "\033[12;15H\c"
read cardseq2
m=1
k=1
;;

9)
echo "\033[13;15H\c"
read cardseq3
m=1
k=1
;;

10)
echo "\033[14;15H\c"
read cardseq4
m=1
k=1
;;

11)
echo "\033[15;15H\c"
read cardseq5
m=1
k=1
;;

12)
echo "\033[16;15H\c"
read cardseq6
m=1
k=1
;;

13)
echo "\033[17;15H\c"
read cardseq7
m=1
k=1
;;

14)
echo "\033[18;15H\c"
read cardseq8
m=1
k=1
;;

*) echo;
;;
```

```
esac
done
##
## END OF ENTRY CHANGE LOOP
##
##
esac
done
##
## END OF MORE CHANGES LOOP
##
##
done
##
## END OF VERIFICATION LOOP
##
```

```
##
## This routine will send an update memo to cfo when
## an action has been taken on an open qdr.
##
clear
##
##  INITIALIZE-THE VARIABLES
##
dmn=" "
ao=" "
apc=" "
actdate=" "
dodaac=" "
action1=" "
action2=" "
##
## THE DATA ENTRY SCREEN
##
echo "\033[2;33HDRS UPDATE ACTION"
echo "\033[3;33H----------------"
echo "\033[5;1HDocument Management Number: \c"
echo "\033[5;45HAction Officer : \c"
echo "\033[7;1HAction Process Code : \c"
echo "\033[7;36HDate of Action (MMDDYY) : \c"
echo "\033[9;1HDODAAC SPT POINT : \c"
echo "\033[11;1HACTION 1 : \c"
echo "\033[5;30H\c"
read dmn
echo "\033[5;62H\c"
read ao
echo "\033[7;23H\c"
read apc
echo "\033[7;62H\c"
read actdate
echo "\033[9;20H\c"
read dodaac
echo "\033[11;12H\c"
read action1
j=0
##
## BEGIN DATA VERIFICATION LOOP
##
while [ "$j" = 0 ]
do
clear
##
## THE DATA SCREEN FOR VERIFICATION
##
echo "\033[2;32HDRS UPDATE MEMO"
echo "\033[3;32H---------------"
echo "\033[5;1H1) DOCUMENT MANAGEMENT NUMBER : $dmn"
echo "\033[5;48H2) ACTION OFFICER : $ao"
```

```
echo "\033[6;33H---------\033[6;68H----"
echo "\033[8;1H3) ACTION PROCESS CODE : $apc"
echo "\033[8;48H4) ACTION DATE : $actdate"
echo "\033[9;26H-\033[9;65H------"
echo "\033[11;1H5) DODAAC SET POINT : $dodaac"
echo "\033[12;23H------"
echo "6) ACTION-1 : $action1"
echo;
k=0
##
## BEGIN MORE CHANGES LOOP
##
while [ "$k" = 0 ]
do
echo "\033[21;20HARE ENTRIES CORRECT (Y/N) ? \c"
read yn
case $yn in
[Yy])
echo "DRS UPDATE ACTION " >> $dmn.$apc
echo "----------------- " >> $dmn.$apc
echo "DOCUMENT MANAGEMENT NUMBER : $dmn \t\tACTION OFFICER : $ao" >> $dmn.
echo " " >> $dmn.$apc
echo "ACTION PROCESS CODE : $apc \t\t\tDATE OF ACTION : $actdate" >> $dmn.$apc
echo " " >> $dmn.$apc
echo "DODAAC SPT POINT : $dodaac >> $dmn.$apc
echo " " >> $dmn.$apc
echo "ACTION 1 : $action1" >> $dmn.$apc
echo " " >> $dmn.$apc
echo "==========================================
mail cfim < $dmn.$apc
echo "\t\t\tSENDING MAIL"
sleep 2
k=1
j=1
;;

[Nn])
m=0
##
## BEGIN ENTRY CHANGE LOOP
##
while [ "$m" = 0 ]
do
echo "\033[21;20HENTER NUMBER TO CHANGE : \c"
read num
```

```
case $num in

1)
echo "\033[5;33H\c"
read dmn
m=1
k=1
;;

2)
echo "\033[5;68H\c"
read ao
m=1
k=1
;;

3)
echo "\033[8;26H\c"
read apc
m=1
k=1
;;

4)
echo "\033[8;65H\c"
read actdate
m=1
k=1
;;

5)
echo "\033[11;23H\c"
read dodaac
m=1
k=1
;;

6)
echo "\033[14;15H\c"
read action1
m=1
k=1
;;

*) echo;
;;
```

```
esac
done
##
## END ENTRY CHANGE LOOP
##
esac
done
##
## END MORE CHANGES LOOP
##
done
##
## END VERIFICATION LOOP
##
```

APPENDIX L

QDR Alerting System

LISTING OF DRS DATABASE TABLES AND DESCRIPTIONS

**** Database: /usr4/acct/cfdba/drsdb/drs ****

action_officers
datetable
last_reply
moraction
reply_due
reply_note_sent
reply_overdue
s2krg0
s2krg1000
s2krg1100
s2krg170
s2krg200
s2krg300
s2krg400
s2krg500
s2krg600
s2krg700
s2krg800
s2krg900

```
****  Database: /usr4/acct/cfdba/drsdb/drs  ****

***  Table: action_officers  ***

    s2kcode                              char(4,1)  Not Null
    name                                 char(20,1)
    userid                               char(8,1)

***  Table: datetable  ***

    today                                date(0)

***  Table: last_reply  ***

    docu_mgt_no                          char(9,3)  Not Null
    last_act_dt                          date(0)

***  Table: moraction  ***

    docu_mgt_no                          char(9,3)  Not Null
    seq_cd_act_procs                     longinteger
    act_procs_cd                         char(1,3)
    act_cd_dt                            date(0)

***  Table: reply_due  ***

    act_off_cd                           char(4,1)
    docu_mgt_no                          char(9,3)
    itmnm_dfp                            char(19,3)
    rec_date                             date(0)
    doc_age                              integer
    last_reply_dt                        date(0)

***  Table: reply_note_sent  ***

    docu_mgt_no                          char(9,3)

***  Table: reply_overdue  ***

    act_off_cd                           char(4,1)
    docu_mgt_no                          char(9,3)
    itmnm_dfp                            char(19,3)
    rec_date                             date(0)
    doc_age                              integer
    last_reply_dt                        date(0)

***  Table: s2krg0  ***
```

docu_mgt_no	char(9,3) Not Null
qdr_eir_rod_wca_cd	char(1,3)
rec_date	date(0)
rept_sta_cd	char(1,3)
rept_sta_cd_dt	date(0)
doc_cat_cd	char(1,3)
dodaac_sor	char(6,3)
adrs_line_sor	text(35,105,35)
src_nme_tel	char(16,3)
rept_con_no	char(12,3)
defn_disc_date	date(0)
nsn_dfp	char(13,3)
add_data_nsn	char(4,3)
act_pt	char(4,3)
fscm_dfp	char(5,3)
mfg_part_no_dfp	char(16,3)
itmnm_dfp	char(19,3)
icp_ric	char(3,3)
acma_ric	char(3,3)
wpn_sys_cd	char(4,3)
secd_prin_item_cd	char(1,3)
dol_val	dollar(9,1)
uprice	dollar(9,1)
ui	char(2,3)
ser_lot_btch_no	char(15,3)
piin	char(13,3)
cao	char(6,3)
dodaac_shipper	char(6,3)
new_rep_ovhl_cd	char(1,3)
new_rep_ovhl_dt	date(0)
op_time_atfal	longinteger
unit_of_quan_msrmt_cd	char(1,3)
gfp	char(1,3)
rec_qnty	longinteger
qnty_insp	longinteger
qnty_defn	longinteger
qty_in_stock	longinteger
est_corr_cst	dollar(6,1)
wrnty	char(1,3)
act_dspo_cd	char(4,3)
cond_cd	char(1,3)
util_cd	char(2,3)
fal_det_dur_cd	char(1,3)
frst_indic_trbl_cd	char(3,3)
fal_cd	char(3,3)
maint_lvl	char(2,3)
ty_eir_qdr_cd	char(4,3)
class_cd	char(2,3)

geo_dpl_area_cd	char(1,3)
act_off_cd	char(4,1)
scrng_pt_nme_tel	char(20,3)
w_nsn	char(13,3)
w_fscm	char(5,3)
w_mfg_part_no	char(16,3)
w_itmnm	char(19,3)
w_ser_no	char(15,3)
orig_uic	char(6,3)
warco_uic	char(6,3)
rep_org_uic	char(6,3)
supp_auto_ltr_cd	char(1,3)
tgt_date	date(0)
w_t_p_cost	dollar(6,1)
w_t_p_cost_clm	dollar(6,1)
t_mhrs_exp	decimal(5,1)
t_mhrs_cost	dollar(6,1)
t_mhrs_cost_clm	dollar(6,1)
t_trans_cost_clm	dollar(6,1)
w_amt_clm	dollar(6,1)
w_amt_reimb	dollar(6,1)
nsn_nha	char(13,3)
add_data_nsn_nha	char(4,3)
fscm_nha	char(5,3)
mfg_part_no_nha	char(16,3)
itmnm_nha	char(19,3)
ser_no_lot_no	char(15,3)
ser_no	char(15,3)
eaa	char(20,3)
func_cd	char(11,3)
nsn_ei	char(13,3)
add_data_nsn_ei	char(4,3)
fscm_ei	char(5,3)
mfg_part_no_ei	char(16,3)
itmnm_ei	char(19,3)
tm_no	char(50,3)
cmd_uniq_1	char(5,3)
cmd_uniq_2	char(5,3)
cmd_uniq_3	char(6,3)
cmd_uniq_4	char(10,3)
cmd_uniq_5	char(25,3)
doc_mgmt_no_dsg	char(1,3)
fast_pay_cd	char(1,3)
src_rec_cd	char(1,3)
ret_whsl_id_cd	char(1,3)
susp_date	date(0)
zip_cd_orig	char(9,3)
fnl_dspn_cd	char(1,3)

```
    cmd_dsg                         char(1,3)
    inv_cat_cd                      char(1,3)
    rqn_no                          char(15,3)
    ty_rptr                         char(1,3)
    ty_shpr                         char(1,3)
    wrng_nsn_rec                    char(13,3)
    date_prp                        date(0)
    icp_ric_fwd                     char(3,3)
    qty_ship                        char(5,3)
    qty_rec_rod                     char(5,3)
    qty_discrp_rod                  char(5,3)
    discrp_tot_cost                 longinteger
    shpr_no                         char(17,3)
    piin_suf                        char(1,3)
    clin                            char(6,3)
    c_o_sn                          char(4,3)
    dodic                           char(4,3)
    proj_cd                         char(3,3)
    blame_cd                        char(4,3)
    loss_cd                         char(1,3)
    act_req                         char(2,3)
    phy_sec_pilferg                 char(1,3)
    dupe_indic_rod                  char(1,3)
    dupe_indic_rod_dt               date(0)

***  Table: s2krg1000  ***

    docu_mgt_no                     char(9,3)  Not Null
    seq_cd_resp_cd                  longinteger
    resp_cd                         char(3,3)
    resp_cd_dt                      date(0)

***  Table: s2krg1100  ***

    docu_mgt_no                     char(9,3)  Not Null
    seq_cd_discrp_cd_rod            longinteger
    discrp_cd_rod                   char(4,3)

***  Table: s2krg170  ***

    docu_mgt_no                     char(9,3)  Not Null
    seq_cd_spt_pt                   longinteger
    dodaac_spt_pt                   char(6,3)
    adrs_line_spt_pt                text(35,105,35)

***  Table: s2krg200  ***

    docu_mgt_no                     char(9,3)  Not Null
```

```
    seq_cd_act_procs                longinteger
    act_procs_cd                    char(1,3)
    act_cd_dt                       date(0)

***  Table: s2krg300  ***

    docu_mgt_no                     char(9,3)  Not Null
    seq_cd_narr_data_dtl            longinteger
    narr_data_dtl                   char(50,3)

***  Table: s2krg400  ***

    docu_mgt_no                     char(9,3)  Not Null
    seq_cd_narr_data_clos           longinteger
    narr_data_clos                  char(50,3)

***  Table: s2krg500  ***

    docu_mgt_no                     char(9,3)  Not Null
    seq_cd_rmks                     longinteger
    rmks                            char(50,3)
    rmks_dt                         date(0)

***  Table: s2krg600  ***

    docu_mgt_no                     char(9,3)  Not Null
    comd_item_dsgr                  char(18,3)

***  Table: s2krg700  ***

    docu_mgt_no                     char(9,3)  Not Null
    ref_desg_cd                     char(1,3)
    ref_desg                        char(60,3)

***  Table: s2krg800  ***

    docu_mgt_no                     char(9,3)  Not Null
    cntr                            char(35,3)
    cntr_adrs                       text(35,140,35)

***  Table: s2krg900  ***

    docu_mgt_no                     char(9,3)  Not Null
    seq_cd_w_disp_cd                longinteger
    w_disp_cd                       char(1,3)
    w_disp_cd_dt                    date(0)
```

LISTING OF FILE *cronjobs*

```
#    CRON JOBS FOR PERIODIC PROCESSING OF DRS DATA

#    Update the DRS database in directory /usr4/acct/cfdba/drsdb
0 20 * * 5 /usr4/acct/cfdba/drsdb/update_drs 1> /usr4/acct/cfdba/
drsdb/drsupdatlog 2>> /usr4/acct/cfdba/drsdb/drsupdaterr

#    Update the DRS table for replies to originator almost overdue
0 2 * * * /usr4/acct/cfdba/drsdb/updt_reply_due 1> /usr4/acct/cfdba/
drsdb/day_reply_log 2>> /usr4/acct/cfdba/drsdb/day_reply_err

#    Update the DRS table for replies to originator already overdue
30 2 * * 1 /usr4/acct/cfdba/drsdb/updt_reply_od 1> /usr4/acct/cfdba/
drsdb/wk_reply_log 2>> /usr4/acct/cfdba/drsdb/wk_reply_err

#    Send 'reply due' notices to Action Officers for replies to
originator
0 4 * * * /usr4/acct/cfdba/drsdb/day_reply_note 1> /usr4/acct/cfdba/
drsdb/reply_due_log 2>> /usr4/acct/cfdba/drsdb/reply_due_err

#    Send weekly 'reply overdue' notices to Action Officers
30 4 * * 1 /usr4/acct/cfdba/drsdb/wk_reply_note 1> /usr4/acct/cfdba/
drsdb/reply_od_log 2>> /usr4/acct/cfdba/drsdb/reply_od_err

#    Send weekly 'reply overdue' notice to Action Officer supervisor
0 5 * * 1 /usr4/acct/cfdba/drsdb/supv_reply_not 1> /usr4/acct/cfdba/
drsdb/supv_reply_log 2>> /usr4/acct/cfdba/drsdb/supv_reply_err

#    Send monthly 'reply overdue' notice to CFO
30 5 1 * * /usr4/acct/cfdba/drsdb/cfo_reply_note 1> /usr4/acct/cfdba/
drsdb/cfo_reply_log 2>> /usr4/acct/cfdba/drsdb/cfo_reply_err
```

LISTING OF FILE *update_drs*

```
# Shellscript to update DRS database

cmd=/usr/bin/mscmd
today=`date`

cd /usr2/acct/tnewby/drs
msclearlock ndrs;
$cmd 'ndrs' "select from s2krg0 dump into '/usr4/acct/cfdba/
drsdb/tab0';"
$cmd 'ndrs' "select from s2krg170 dump into '/usr4/acct/cfdba/
drsdb/tab170';"
$cmd 'ndrs' "select from s2krg200 dump into '/usr4/acct/cfdba/
drsdb/tab200';"
$cmd 'ndrs' "select from s2krg300 dump into '/usr4/acct/cfdba/
drsdb/tab300';"
$cmd 'ndrs' "select from s2krg400 dump into '/usr4/acct/cfdba/
drsdb/tab400';"
$cmd 'ndrs' "select from s2krg500 dump into '/usr4/acct/cfdba/
drsdb/tab500';"
$cmd 'ndrs' "select from s2krg600 dump into '/usr4/acct/cfdba/
drsdb/tab600';"
$cmd 'ndrs' "select from s2krg700 dump into '/usr4/acct/cfdba/
drsdb/tab700';"
$cmd 'ndrs' "select from s2krg800 dump into '/usr4/acct/cfdba/
drsdb/tab800';"
$cmd 'ndrs' "select from s2krg900 dump into '/usr4/acct/cfdba/
drsdb/tab900';"
$cmd 'ndrs' "select from s2krg1000 dump into '/usr4/acct/cfdba/
drsdb/tab1000';"
$cmd 'ndrs' "select from s2krg1100 dump into '/usr4/acct/cfdba/
drsdb/tab1100';"
msclearlock  ndrs;

cd /usr4/acct/cfdba/drsdb
msclearlock drs;
rm updatrcd
$cmd 'drs' "drop index dmn;"
$cmd 'drs' "drop index cur_status;"
$cmd 'drs' "drop index stock_nr;"
$cmd 'drs' "drop index wpn_sys;"
$cmd 'drs' "drop index contr_nr;"
$cmd 'drs' "drop index act_off;"
$cmd 'drs' "empty s2krg0;"
$cmd 'drs' "alter s2krg0 change act_off_cd char (4, 3);"
```

```
$cmd 'drs' "insert into s2krg0 from 'tab0';"
$cmd 'drs' "update s2krg0 set act_off_cd to null where act_off_cd match
'    ';"
$cmd 'drs' "alter s2krg0 change act_off_cd char (4, 1);"
$cmd 'drs' "create unique index dmn on s2krg0 (docu_mgt_no);"
$cmd 'drs' "create index cur_status on s2krg0 (rept_sta_cd);"
$cmd 'drs' "create index stock_nr on s2krg0 (nsn_dfp);"
$cmd 'drs' "create index wpn_sys on s2krg0 (wpn_sys_cd);"
$cmd 'drs' "create index contr_nr on s2krg0 (piin);"
$cmd 'drs' "create index act_off on s2krg0 (act_off_cd);"
rm tab0
echo Table s2krg0 updated on $today > updatrcd

$cmd 'drs' "drop index dmn170;"
$cmd 'drs' "drop index spt_pt;"
$cmd 'drs' "empty s2krg170;"
$cmd 'drs' "insert into s2krg170 from 'tab170';"
$cmd 'drs' "create index dmn170 on s2krg170 (docu_mgt_no);"
$cmd 'drs' "create index spt_pt on s2krg170 (dodaac_spt_pt);"
rm tab170
echo Table s2krg170 updated on $today >> updatrcd

$cmd 'drs' "drop index dmn200;"
$cmd 'drs' "drop index act_code;"
$cmd 'drs' "empty s2krg200;"
$cmd 'drs' "insert into s2krg200 from 'tab200';"
$cmd 'drs' "create index dmn200 on s2krg200 (docu_mgt_no);"
$cmd 'drs' "create index act_code on s2krg200 (act_procs_cd);"
rm tab200
echo Table s2krg200 updated on $today >> updatrcd

$cmd 'drs' "drop index dmn300;"
$cmd 'drs' "empty s2krg300;"
$cmd 'drs' "insert into s2krg300 from 'tab300';"
$cmd 'drs' "create index dmn300 on s2krg300 (docu_mgt_no);"
rm tab300
echo Table s2krg300 updated on $today >> updatrcd

$cmd 'drs' "drop index dmn400;"
$cmd 'drs' "empty s2krg400;"
$cmd 'drs' "insert into s2krg400 from 'tab400';"
$cmd 'drs' "create index dmn400 on s2krg400 (docu_mgt_no);"
rm tab400
echo Table s2krg400 updated on $today >> updatrcd

$cmd 'drs' "drop index dmn500;"
$cmd 'drs' "empty s2krg500;"
$cmd 'drs' "insert into s2krg500 from 'tab500';"
```

```
$cmd 'drs' "create index dmn500 on s2krg500 (docu_mgt_no);"
rm tab500
echo Table s2krg500 updated on $today >> updatrcd

$cmd 'drs' "drop index dmn600;"
$cmd 'drs' "drop index cmd_item;"
$cmd 'drs' "empty s2krg600;"
$cmd 'drs' "insert into s2krg600 from 'tab600';"
$cmd 'drs' "create index dmn600 on s2krg600 (docu_mgt_no);"
$cmd 'drs' "create index cmd_item on s2krg600 (comd_item_dsgr);"
rm tab600
echo Table s2krg600 updated on $today >> updatrcd

$cmd 'drs' "drop index dmn700;"
$cmd 'drs' "drop index electr_ref;"
$cmd 'drs' "empty s2krg700;"
$cmd 'drs' "insert into s2krg700 from 'tab700';"
$cmd 'drs' "create index dmn700 on s2krg700 (docu_mgt_no);"
$cmd 'drs' "create index electr_ref on s2krg700 (ref_desg_cd);"
rm tab700
echo Table s2krg700 updated on $today >> updatrcd

$cmd 'drs' "drop index dmn800;"
$cmd 'drs' "drop index contractor;"
$cmd 'drs' "empty s2krg800;"
$cmd 'drs' "insert into s2krg800 from 'tab800';"
$cmd 'drs' "create index dmn800 on s2krg800 (docu_mgt_no);"
$cmd 'drs' "create index contractor on s2krg800 (cntr);"
rm tab800
echo Table s2krg800 updated on $today >> updatrcd

$cmd 'drs' "drop index dmn900;"
$cmd 'drs' "drop index war_disp;"
$cmd 'drs' "empty s2krg900;"
$cmd 'drs' "insert into s2krg900 from 'tab900';"
$cmd 'drs' "create index dmn900 on s2krg900 (docu_mgt_no);"
$cmd 'drs' "create index war_disp on s2krg900 (w_disp_cd);"
rm tab900
echo Table s2krg900 updated on $today >> updatrcd

$cmd 'drs' "drop index dmn1000;"
$cmd 'drs' "drop index resp;"
$cmd 'drs' "empty s2krg1000;"
$cmd 'drs' "insert into s2krg1000 from 'tab1000';"
$cmd 'drs' "create index dmn1000 on s2krg1000 (docu_mgt_no);"
$cmd 'drs' "create index resp on s2krg1000 (resp_cd);"
rm tab1000
echo Table s2krg1000 updated on $today >> updatrcd
```

```
$cmd 'drs' "drop index dmn1100;"
$cmd 'drs' "drop index discrep;"
$cmd 'drs' "empty s2krg1100;"
$cmd 'drs' "insert into s2krg1100 from 'tab1100';"
$cmd 'drs' "create index dmn1100 on s2krg1100 (docu_mgt_no);"
$cmd 'drs' "create index discrep on s2krg1100 (discrp_cd_rod);"
rm tab1100
echo Table s2krg1100 updated on $today >> updatrcd

msclearlock drs;
exit
```

LISTING OF FILE *updt_reply_due*

```
# Shellscript to update the table 'reply_due' in the DRS database
# This table contains a list of the QDRs which will become overdue
# in the next two weeks and will require a reply to the QDR
originator

cmd=/usr/bin/mscmd
today=`date`

cd /usr4/acct/cfdba/drsdb
msclearlock drs;

$cmd 'drs' "run updt_day_reply;"

msclearlock drs;
exit
```

LISTING OF FILE *updt_reply_od*

```
# Shellscript to update the table 'reply_overdue' in the DRS database
# This table contains a list of the QDRs which are already overdue
# and now require a reply to the QDR originator

cmd=/usr/bin/mscmd
today=`date`

cd /usr4/acct/cfdba/drsdb
msclearlock drs;

$cmd 'drs' "run updt_wk_reply;"

msclearlock drs;
exit
```

LISTING OF FILE *updt_day_reply*

```
/* Routine to update the table which contains data to notify
   the Action Officers that replies to the QDR originator are due */


update datetable set today = 'today';

empty last_reply;

select distinct docu_mgt_no, max (act_cd_dt) from s2krg200
where act_procs_cd match '[BCV]'
group by docu_mgt_no
insert into last_reply;

empty reply_due;

      /*  Add documents where no action has been reported yet */

select act_off_cd, docu_mgt_no, itmnm_dfp, rec_date, (today -
rec_date)
  from s2krg0, datetable

where     rept_sta_cd = 'O'

and       act_off_cd match (select s2kcode from action_officers)

and       (rec_date + 90 days) between today and (today + 2 weeks)

and       not exists (select docu_mgt_no from s2krg200
              where s2krg200.docu_mgt_no = s2krg0.docu_mgt_no)

and       not exists (select docu_mgt_no from reply_note_sent
              where reply_note_sent.docu_mgt_no = s2krg0.docu_mgt_no)

insert into reply_due;


      /*  Add documents where action has been reported but the
    QDR originator has not been notified yet */

select act_off_cd, docu_mgt_no, itmnm_dfp , rec_date, (today -
rec_date)
  from s2krg0, datetable

where     rept_sta_cd = 'O'
```

```
and       act_off_cd match (select s2kcode from action_officers)

and       (rec_date + 90 days) between today and (today + 2 weeks)

and       exists (select docu_mgt_no from s2krg200
          where s2krg200.docu_mgt_no = s2krg0.docu_mgt_no)

and       not exists (select docu_mgt_no from s2krg200
              where s2krg200.docu_mgt_no = s2krg0.docu_mgt_no
              and act_procs_cd match '[BCV]')

and       not exists (select docu_mgt_no from reply_note_sent
              where reply_note_sent.docu_mgt_no = s2krg0.docu_mgt_no)

insert into reply_due;


      /*  Add documents where action has been reported but the
    QDR originator was notified over 60 days ago */

select act_off_cd, docu_mgt_no, itmnm_dfp, rec_date,
  (today - rec_date), last_act_dt
  from s2krg0, last_reply, datetable

where     rept_sta_cd = 'O'

and       act_off_cd match (select s2kcode from action_officers)


and       (last_reply.docu_mgt_no = s2krg0.docu_mgt_no
  and (last_act_dt + 60 days) between today and (today + 2 weeks))

and       not exists (select docu_mgt_no from reply_note_sent
              where reply_note_sent.docu_mgt_no = s2krg0.docu_mgt_no)

insert into reply_due;
```

LISTING OF FILE *updt_wk_reply*

```
/* Routine to update the table which contains data to notify the
   Action Officers that replies to the QDR originator are overdue */


update datetable set today = 'today';

empty last_reply;

select distinct docu_mgt_no, max (act_cd_dt) from s2krg200 where
act_procs_cd match '[BCV]'
group by docu_mgt_no
insert into last_reply;

empty reply_overdue;

      /*  Add documents where no action has been reported yet */

select act_off_cd, docu_mgt_no, itmnm_dfp, rec_date, (today -
rec_date)
  from s2krg0, datetable

where     rept_sta_cd = 'O' and (rec_date + 90 days + 2 weeks) <=
today

and       act_off_cd match (select s2kcode from action_officers)

and       not exists (select docu_mgt_no from s2krg200
              where s2krg200.docu_mgt_no = s2krg0.docu_mgt_no)

insert into reply_overdue;


      /*  Add documents where action has been reported but the
    QDR originator has not been notified yet */

select act_off_cd, docu_mgt_no, itmnm_dfp , rec_date, (today -
rec_date)
  from s2krg0, datetable

where     rept_sta_cd = 'O' and (rec_date + 90 days + 2 weeks) <=
today

and       act_off_cd match (select s2kcode from action_officers)
```

```
and         exists (select docu_mgt_no from s2krg200
            where s2krg200.docu_mgt_no = s2krg0.docu_mgt_no)

and         not exists (select docu_mgt_no from s2krg200
                where s2krg200.docu_mgt_no = s2krg0.docu_mgt_no
                and act_procs_cd match '[BCV]')

insert into reply_overdue;


      /*  Add documents where action has been reported but the
    QDR originator was notified over 60 days ago */

select act_off_cd, docu_mgt_no, itmnm_dfp, rec_date,
  (today - rec_date), last_act_dt
  from s2krg0, last_reply, datetable

where       rept_sta_cd = 'O' and (rec_date + 90 days + 2 weeks) <=
today

and         act_off_cd match (select s2kcode from action_officers)

and         (last_reply.docu_mgt_no = s2krg0.docu_mgt_no
  and (last_act_dt + 60 days + 2 weeks) <= today)

insert into reply_overdue;
```

LISTING OF FILE day_reply_note

```
#    This shellscript will search the reply due file (for replies to
#    the QDR originator) and send notices to each Action Officer if
required


cmd=/usr/bin/mscmd

#    Send notices to action officers as required

officers='BJ EA GSR JC LMB MJ RT SJS'

for ao in $officers
do
  $cmd 'drs' "select count from reply_due dump into reply_cnt where
act_off_cd match '$ao';"
  qty_due=`cat reply_cnt`

  if [ "$qty_due" -gt "0" ]; then
          case $ao in
                  BJ)     name='Billy James'
                          userid=bjames ;;

                  EA)     name='Emmett Ashworth'
                          userid=eashwort ;;

                  GSR)    name='Gadson Ross'
                          userid=gross ;;

                  JC)     name='Joyce Cooper'
                          userid=jcooper ;;

                  LMB)    name='Lynn Blanton'
                          userid=wblanton ;;

                  MJ)     name='Milton Jones'
                          userid=mjones ;;

                  RT)     name='Ron Thomas'
                          userid=rthomas ;;

                  SJS)    name='Shelby Scott'
                          userid=sscott ;;

                  *)      break ;;
```

```
            esac

            mwriter reply_due_rpt "$ao" "$name" > tempfile
            mail '$userid' < tempfile
            $cmd 'drs' "select docu_mgt_no from reply_due where
act_off_cd match '$ao' insert into reply_note_sent;"
  fi
done
exit
```

LISTING OF FILE *wk_reply_note*

```
#    This shellscript will search the reply overdue file (for replies
#    to the QDR originator) and send notices to each Action Officer
if required


cmd=/usr/bin/mscmd

#    Send notices to action officers as required

officers='BJ EA GSR JC LMB MJ RT SJS'

for ao in $officers
do
  $cmd 'drs' "select count from reply_overdue dump into reply_cnt
where act_off_cd match '$ao';"
  qty_due=`cat reply_cnt`

  if [ "$qty_due" -gt "0" ]; then
          case $ao in
                  BJ)     name='Billy James'
                          userid=bjames ;;

                  EA)     name='Emmett Ashworth'
                          userid=eashwort ;;

                  GSR)    name='Gadson Ross'
                          userid=gross ;;

                  JC)     name='Joyce Cooper'
                          userid=jcooper ;;

                  LMB)    name='Lynn Blanton'
                          userid=wblanton ;;

                  MJ)     name='Milton Jones'
                          userid=mjones ;;

                  RT)     name='Ron Thomas'
                          userid=rthomas ;;

                  SJS)    name='Shelby Scott'
                          userid=sscott ;;

                  *)      break ;;
```

```
            esac

            mwriter reply_od_rpt "$ao" "$name" > tempfile
            mail '$userid' < tempfile
        fi
    done
    exit
```

LISTING OF FILE *supv_reply_not*

```
#     This shellscript will search the reply overdue file (for replies
#     to the QDR originator) and send the Action Officer supervisor a
#     list of QDRs requiring reply to the originator


cmd=/usr/bin/mscmd
userid=fowens

$cmd 'drs' "select count from reply_overdue dump into reply_cnt;"
qty_due=`cat reply_cnt`

if [ "$qty_due" -gt "0" ]; then
  mwriter supv_reply_rpt > tempfile
  mail '$userid' < tempfile
fi
rm tempfile
exit
```

LISTING OF FILE *cfo_reply_note*

```
#     This shellscript will search the reply overdue file (for replies
#     to the QDR originator) and send CFO a list of QDRs requiring
#     reply to the originator


cmd=/usr/bin/mscmd
userid=msyancey

$cmd 'drs' "select count from reply_overdue dump into reply_cnt;"
qty_due=`cat reply_cnt`

if [ "$qty_due" -gt "0" ]; then
  mwriter cfo_reply_rpt > tempfile
  mail '$userid' < tempfile
fi
rm tempfile
exit
```

LISTING OF FILE *reply_due_rpt*

```
/*      Program to print list of documents where interim/final  */
/*      reply to originator will be due in the next two weeks   */

parameters
        char    ao_code;
        char    ao_name;
end;

let DATABASE = "drs";
let PAGEWIDTH = 80;
let boxbottom = 50;

let boxtop =
 "__________________________________________________________________
____";
let boxhead =
 "|-----------------------------------------------------------------
---|";
let boxline =
 "|            |                     |           |         |
|";
let boxmid =
 "|------------|---------------------|-----------|---------|---------
---|";
let boxend =
 "|____________|_____________________|___________|_________|_________
___|";
let boxbase =
 "|_________________________________________________________________
___|";

header one
        newline 2;
        print col 5, "Memo to: ", col 14, var.ao_name, col 62, TODAY
picture "dd aaa yy" width 9 right, newline 2;
        print col 5, "From: DRS Database Administrator", newline 2;
        print col 5, "Subject: QDRs Due for Reply to QDR Originator
in Next Two Weeks", newline 3;
        print col 9, "According to the current contents of the DRS
database on the", newline;
        print col 5, "Unisys 5000 computer, the following documents
will require an", newline;
        print col 5, "interim or final reply to the QDR originator in
the next two weeks.",newline;
```

```
        print col 5, "If this listing is incorrect, please advise CFO
so that the",newline;
        print col 5, "database may be updated.", newline;

        newline 3;
        print col 6, boxtop, newline;
        print col 6,"|  DOCUMENT  |", col 41, "|   DATE    | CURRENT
|",
                col 76, "|", newline;
        print col 6, "| MANAGEMENT |", col 41, "| DOCUMENT  |   AGE
|",
                col 65, "DATE LAST  |", newline;
        print col 6, "|     NR     |       ITEM NAME       | RECEIVED
|",
                 col 56, "(DAYS) | REPLY SENT |", newline;
        print col 6, boxmid, newline;
end;

header two
        newline 2;
        print col 6, boxtop, newline;
        print col 6, "|", col 27,"Continuation of QDR Listing",
                        col 76, "|", newline;
        print col 6, boxhead, newline;
        print col 6,"|  DOCUMENT  |", col 41, "|   DATE    | CURRENT
|",
                col 76, "|", newline;
        print col 6, "| MANAGEMENT |", col 41, "| DOCUMENT  |   AGE
|",
                col 65, "DATE LAST  |", newline;
        print col 6, "|     NR     |       ITEM NAME       | RECEIVED
|",
                 col 56, "(DAYS) | REPLY SENT |", newline;
        print col 6, boxmid, newline;
end;

select act_off_cd, docu_mgt_no, itmnm_dfp, rec_date, doc_age,
last_reply_dt
        from reply_due where act_off_cd match 'var.ao_code'
        sort by docu_mgt_no;

for each group of act_off_cd
        use header one;

        for each record
                print col 6, "|", col 9, docu_mgt_no,
                        col 19, "|", col 21, itmnm_dfp,
                        col 41, "|", col 43,rec_date picture "dd
```

```
aaa yy"
                                width 9 right,
                                col 53, "|", col 55, doc_age picture
"z@,zz9",
                                col 63, "|", col 66, last_reply_dt picture
"dd aaa yy"
                                width 9 right,
                                col 76, "|", newline;
                        if LINENUMBER > boxbottom
                                then
                                print col 6, boxend, newline;
                                print col 6, "|", col 76, "|", newline;
                                print col 6,"|", col 30, "CONTINUED ON NEXT
PAGE",
                                        col 76, "|", newline;
                                print col 6, boxbase, newline;
                                if PAGENUMBER > 1
                                        then
                                        newline 2;
                                        print col 38, "Page ",
                                                PAGENUMBER picture "Z",
newline;
                                end;
                                use header two;
                                newpage;
                        end;
                end;

                print col 6, boxline, newline, col 6, boxend, newline;
                let nr_docs = count docu_mgt_no;
                print col 6, "|", col 76, "|", newline;
                print col 6, "|", col 25, "TOTAL NUMBER OF DOCUMENTS =",
                        nr_docs picture "ZZ9", col 76, "|", newline;
                print col 6, boxbase, newline;
                if PAGENUMBER > 1
                        then
                        newline 2;
                        print col 38, "Page ",
                                PAGENUMBER picture "Z", newline;
                end;
                newpage;
                let PAGENUMBER = 1;
end;
```

LISTING OF FILE *reply_od_rpt*

```
/*      Program to print list of documents where interim/final  */
/*      reply to originator is overdue                          */

parameters
        char    ao_code;
        char    ao_name;
end;

let DATABASE = "drs";
let PAGEWIDTH = 80;
let boxbottom = 50;

let boxtop =
 "
 ";
let boxhead =
 "|
-----|";
let boxline =
 "|            |                    |          |         |
|";
let boxmid =
 "|------------|--------------------|----------|---------|-------
-----|";
let boxend =
 "|____________|____________________|__________|_________|_______
_____|";
let boxbase =
 "|
_____|";

header one
        newline 2;
        print col 5, "Memo to: ", col 14, var.ao_name, col 62,
TODAY picture "dd aaa yy" width 9 right, newline 2;
        print col 5, "From: DRS Database Administrator", newline 2;
        print col 5, "Subject: QDRs Overdue for Reply to QDR
Originator", newline 3;
        print col 9, "According to the current contents of the DRS
database on the", newline;
        print col 5, "Unisys 5000 computer, the following documents
now require an interim", newline;
        print col 5, "or final reply to the QDR originator.  If
this listing is incorrect,", newline;
```

```
        print col 5, "please advise CFO so that the database may be
updated.", newline;

        newline 3;
        print col 6, boxtop, newline;
        print col 6,"|  DOCUMENT  |", col 41, "|   DATE    |
CURRENT |",
                col 76, "|", newline;
        print col 6, "| MANAGEMENT |", col 41, "| DOCUMENT  |   AGE
|",
                col 65, "DATE LAST  |", newline;
        print col 6, "|     NR     |       ITEM NAME       | RECEIVED
|",
                 col 56, "(DAYS) | REPLY SENT |", newline;
        print col 6, boxmid, newline;
end;

header two
        newline 2;
        print col 6, boxtop, newline;
        print col 6, "|", col 27,"Continuation of QDR Listing",
                        col 76, "|", newline;
        print col 6, boxhead, newline;
        print col 6,"|  DOCUMENT  |", col 41, "|   DATE    |
CURRENT |",
                col 76, "|", newline;
        print col 6, "| MANAGEMENT |", col 41, "| DOCUMENT  |   AGE
|",
                col 65, "DATE LAST  |", newline;
        print col 6, "|     NR     |       ITEM NAME       | RECEIVED
|",
                 col 56, "(DAYS) | REPLY SENT |", newline;
        print col 6, boxmid, newline;
end;

select act_off_cd, docu_mgt_no, itmnm_dfp, rec_date, doc_age,
last_reply_dt
        from reply_due where act_off_cd match 'var.ao_code'
        sort by docu_mgt_no;

for each group of act_off_cd
        use header one;

        for each record
                print col 6, "|", col 9, docu_mgt_no,
                        col 19, "|", col 21, itmnm_dfp,
                        col 41, "|", col 43,rec_date picture "dd
aaa yy"
```

```
                        width 9 right,
                        col 53, "|", col 55, doc_age picture
"Z@,ZZ9",
                        col 63, "|", col 66, last_reply_dt picture
"dd aaa yy"
                        width 9 right,
                        col 76, "|", newline;
                if LINENUMBER > boxbottom
                        then
                        print col 6, boxend, newline;
                        print col 6, "|", col 76, "|", newline;
                        print col 6,"|", col 30, "CONTINUED ON NEXT
PAGE",
                                col 76, "|", newline;
                        print col 6, boxbase, newline;
                        if PAGENUMBER > 1
                                then
                                newline 2;
                                print col 38, "Page ",
                                        PAGENUMBER picture "Z",
newline;
                        end;
                        use header two;
                        newpage;
                end;
        end;

        while LINENUMBER < boxbottom
                do
                print col 6, boxline, newline;
        end;

        print col 6, boxline, newline, col 6, boxend, newline;
        let nr_docs = count docu_mgt_no;
        print col 6, "|", col 76, "|", newline;
        print col 6, "|", col 25, "TOTAL NUMBER OF DOCUMENTS =",
                nr_docs picture "ZZ9", col 76, "|", newline;
        print col 6, boxbase, newline;
        if PAGENUMBER > 1
                then
                newline 2;
                print col 38, "Page ",
                        PAGENUMBER picture "Z", newline;
        end;
        newpage;
        let PAGENUMBER = 1;
end;
```

LISTING OF FILE *supv_reply_rpt*

```
/*      Program to print list of all documents where interim or  */
/*      final reply to originator is overdue                     */

let DATABASE = "drs";

select act_off_cd, docu_mgt_no, itmnm_dfp, rec_date, doc_age,
last_reply_dt
  from reply_overdue sort by act_off_cd, docu_mgt_no;

let PAGEWIDTH = 80;
let boxbottom = 50;

let boxtop =
  "
__";
let boxhead =
  "|-------------------------------------------------------------------
-|";
let boxline =
  "|             |                      |           |         |
|";
let boxmid =
  "|-------------|----------------------|-----------|---------|----------
-|";
let boxend =
  "|_____________|______________________|___________|_________|_________
_|";
let boxbase =
  "|_____________________________________________________________________
_|";

header one
  newline 2;
  print col 5, "Memo to: Chief, Qual Rqmnts and Data Eval Branch", col
62, TODAY picture "dd aaa yy" width 9 right, newline 2;
  print col 5, "From: DRS Database Administrator", newline 2;
  print col 5, "Subject: QDRs Overdue for Reply to QDR Originator",
newline 3;
  print col 9, "According to the current contents of the DRS database
on the", newline;
  print col 5, "Unisys 5000 computer, the following documents now
require an interim", newline;
  print col 5, "or final reply to the QDR originator.  If this listing
is incorrect,", newline;
```

```
  print col 5, "please advise CFO so that the database may be
updated.", newline;
end;

header two
  newline 2;
  print col 6, boxtop, newline;
  print col 6, "|", col 27,"Continuation of QDR Listing",
                  col 76, "|", newline;
  print col 6, boxhead, newline;
  print col 6, "|  Action Officer Code: ", col 30, act_off_cd,
          col 76, "|",newline;
  print col 6, boxhead, newline;
  print col 6,"|  DOCUMENT  |", col 41, "|   DATE    | CURRENT |",
          col 76, "|", newline;
  print col 6, "| MANAGEMENT |", col 41, "| DOCUMENT  |   AGE   |",
          col 65, "DATE LAST  |", newline;
  print col 6, "|     NR     |      ITEM NAME      | RECEIVED  |",
           col 56, "(DAYS) | REPLY SENT |", newline;
  print col 6, boxmid, newline;
end;

use header one;

for each group of act_off_cd

  newline 2;
  print col 6, boxtop, newline;
  print col 6, "|  Action Officer Code: ", col 30, act_off_cd,
          col 76, "|",newline;
  print col 6, boxhead, newline;
  print col 6,"|  DOCUMENT  |", col 41, "|   DATE    | CURRENT |",
          col 76, "|", newline;
  print col 6, "| MANAGEMENT |", col 41, "| DOCUMENT  |   AGE   |",
          col 65, "DATE LAST  |", newline;
  print col 6, "|     NR     |      ITEM NAME      | RECEIVED  |",
           col 56, "(DAYS) | REPLY SENT |", newline;
  print col 6, boxmid, newline;

  for each record
          print col 6, "|", col 9, docu_mgt_no,
                  col 19, "|", col 21, itmnm_dfp,
                  col 41, "|", col 43,rec_date picture "dd aaa yy"
                  width 9 right,
                  col 53, "|", col 55, doc_age picture "Z@,ZZ9",
                  col 63, "|", col 66, last_reply_dt picture "dd aaa
yy"
                  width 9 right,
```

```
                col 76, "|", newline;
        if LINENUMBER > boxbottom
                then
                print col 6, boxend, newline;
                print col 6, "|", col 76, "|", newline;
                print col 6,"|", col 30, "CONTINUED ON NEXT PAGE",
                        col 76, "|", newline;
                print col 6, boxbase, newline;
                if PAGENUMBER > 1
                        then
                        newline 2;
                        print col 38, "Page ",
                                PAGENUMBER picture "ZZ", newline;
                end;
                use header two;
                newpage;
        end;
  end;

  print col 6, boxline, newline, col 6, boxend, newline;
  let nr_docs = count docu_mgt_no;
  print col 6, "|", col 76, "|", newline;
  print col 6, "|", col 25, "TOTAL NUMBER OF DOCUMENTS =",
        nr_docs picture "ZZ9", col 76, "|", newline;
  print col 6, boxbase, newline;
end;

if PAGENUMBER > 1
  then
  newline 2;
  print col 38, "Page ", PAGENUMBER picture "ZZ", newline;
end;
```

LISTING OF FILE *cfo_reply_rpt*

```
/*      Program to provide CFO a list of documents where interim  */
/*      or final reply to originator is overdue                    */

let DATABASE = "drs";
let PAGELENGTH = 57;
let PAGEWIDTH = 80;
let boxbottom = 49;

let boxtop =
  "
__";
let boxhead =
  "|-------------------------------------------------------------------
-|";
let boxline =
  "|             |                       |           |         |
|";
let boxmid =
  "|-------------|-----------------------|-----------|---------|-----------
-|";
let boxend =
  "|_____________|_______________________|___________|_________|___________
_|";
let boxbase =
  "|_____________________________________________________________________
_|";

header one
  newline 4;
  print col 5, "Memo to: Chief, Customer Feedback Office", col 62,
TODAY picture "dd aaa yy" width 9 right, newline 2;
  print col 5, "From: DRS Database Administrator", newline 2;
  print col 5, "Subject: QDRs Overdue for Reply to QDR Originator",
newline 3;
  print col 9, "According to the current contents of the DRS database
on the", newline;
  print col 5, "Unisys 5000 computer, the following documents now
require an interim", newline;
  print col 5, "or final reply to the QDR originator.  If this listing
is incorrect,", newline;
  print col 5, "please advise so that the database can be updated.",
newline;

  newline 3;
```

```
  print col 6, boxtop, newline;
  print col 6,"|  DOCUMENT  |", col 41, "|   DATE    | CURRENT |",
        col 76, "|", newline;
  print col 6, "| MANAGEMENT |", col 41, "| DOCUMENT  |   AGE   |",
        col 65, "DATE LAST  |", newline;
  print col 6, "|     NR     |      ITEM NAME      | RECEIVED  |",
         col 56, "(DAYS) | REPLY SENT |", newline;
  print col 6, boxmid, newline;
end;

header two
  newline 4;
  print col 6, boxtop, newline;
  print col 6, "|", col 27,"Continuation of QDR Listing",
                col 76, "|", newline;
  print col 6, boxhead, newline;
  print col 6,"|  DOCUMENT  |", col 41, "|   DATE    | CURRENT |",
        col 76, "|", newline;
  print col 6, "| MANAGEMENT |", col 41, "| DOCUMENT  |   AGE   |",
        col 65, "DATE LAST  |", newline;
  print col 6, "|     NR     |      ITEM NAME      | RECEIVED  |",
         col 56, "(DAYS) | REPLY SENT |", newline;
  print col 6, boxmid, newline;
end;

select docu_mgt_no, itmnm_dfp, rec_date, doc_age, last_reply_dt
  from reply_overdue sort by docu_mgt_no;

use header one;

for each record
  print col 6, "|", col 9, docu_mgt_no,
        col 19, "|", col 21, itmnm_dfp,
        col 41, "|", col 43,rec_date picture "dd aaa yy"
        width 9 right,
        col 53, "|", col 55, doc_age picture "Z@,ZZ9",
        col 63, "|", col 66, last_reply_dt picture "dd aaa yy"
        width 9 right,
        col 76, "|", newline;
  if LINENUMBER > boxbottom
        then
        print col 6, boxend, newline;
        print col 6, "|", col 76, "|", newline;
        print col 6,"|", col 30, "CONTINUED ON NEXT PAGE",
                col 76, "|", newline;
        print col 6, boxbase, newline;
        if PAGENUMBER > 1
                then
```

```
                    newline 2;
                    print col 38, "Page ", PAGENUMBER picture "ZZ",
newline;
          end;
          use header two;
          newpage;
  end;
end;

while LINENUMBER < boxbottom
  do
  print col 6, boxline, newline;
end;

print col 6, boxline, newline, col 6, boxend, newline;
let nr_docs = count docu_mgt_no;
print col 6, "|", col 76, "|", newline;
print col 6, "|", col 25, "TOTAL NUMBER OF DOCUMENTS =",
  nr_docs picture "ZZ9", col 76, "|", newline;
print col 6, boxbase, newline;
if PAGENUMBER > 1
  then
  newline 2;
  print col 38, "Page ", PAGENUMBER picture "ZZ", newline;
end;
```

EXAMPLE OF DAILY NOTICE TO ACTION OFFICERS

Memo to: John Doe (Action Officer) 20 Sep 89

From: DRS Database Administrator

Subject: QDRs Due for Reply to QDR Originator in Next Two Weeks

According to the current contents of the DRS database on the Unisys 5000 computer, the following documents will require an interim or final reply to the QDR originator in the next two weeks. If this listing is incorrect, please advise CFO so that the database may be updated.

DOCUMENT MANAGEMENT NR	ITEM NAME	DATE DOCUMENT RECEIVED	CURRENT AGE (DAYS)	DATE LAST REPLY SENT
P19L00514	CABLE ASSY, RADI	22 Jun 89	90	
P19L00523	(LEM)STRUCTURE ASSY	6 Jul 89	76	
P19L00530	ACTUATOR,EXPLOSIVE,	6 Jul 89	76	
P19L00531	CABLE ASSEMBLY,SPEC	6 Jul 89	76	
TOTAL NUMBER OF DOCUMENTS = 4				

EXAMPLE OF WEEKLY NOTICE TO ACTION OFFICERS

Memo to: John Doe (Action Officer) 20 Sep 89

From: DRS Database Administrator

Subject: QDRs Overdue for Reply to QDR Originator

According to the current contents of the DRS database on the Unisys 5000 computer, the following documents now require an interim or final reply to the QDR originator. If this listing is incorrect, please advise CFO so that the database may be updated.

DOCUMENT MANAGEMENT NR	ITEM NAME	DATE DOCUMENT RECEIVED	CURRENT AGE (DAYS)	DATE LAST REPLY SENT
P19L00442	CABLE AND SOLENOID	5 Jun 89	107	
P19L00443	LATCH ASSEMBLY,DOOR	5 Jun 89	107	
TOTAL NUMBER OF DOCUMENTS = 2				

EXAMPLE OF WEEKLY NOTICE TO ACTION OFFICER SUPERVISOR

Memo to: Chief, Qual Rqmnts and Data Eval Branch 20 Sep 89

From: DRS Database Administrator

Subject: QDRs Overdue for Reply to QDR Originator

According to the current contents of the DRS database on the Unisys 5000 computer, the following documents now require an interim or final reply to the QDR originator. If this listing is incorrect, please advise CFO so that the database may be updated.

Action Officer Code: AO1

DOCUMENT MANAGEMENT NR	ITEM NAME	DATE DOCUMENT RECEIVED	CURRENT AGE (DAYS)	DATE LAST REPLY SENT
P19L00428	SHELL,OUTER,BODY SE	26 May 89	117	

TOTAL NUMBER OF DOCUMENTS = 1

Action Officer Code: AO2

DOCUMENT MANAGEMENT NR	ITEM NAME	DATE DOCUMENT RECEIVED	CURRENT AGE (DAYS)	DATE LAST REPLY SENT
P19L00442	CABLE AND SOLENOID	5 Jun 89	107	
P19L00443	LATCH ASSEMBLY,DOOR	5 Jun 89	107	

TOTAL NUMBER OF DOCUMENTS = 2

Action Officer Code: AO3

DOCUMENT MANAGEMENT NR	ITEM NAME	DATE DOCUMENT RECEIVED	CURRENT AGE (DAYS)	DATE LAST REPLY SENT
P18L00034	CIRCUIT CARD ASSEMB	20 Jan 88	609	17 Nov 88
P18L00059	CIRCUIT CARD ASSEMB	29 Jan 88	600	17 Nov 88
P18L00335	MODIFICATION KIT	21 Apr 88	517	
P18L00607	LOADER-TRANSPORTER,	27 Jul 88	420	7 Mar 89
P18L00693	CABLE ASSEMBLY,SPEC	7 Sep 88	378	1 Dec 88
P18L00700	GUIDED MISSILE,TRAI	8 Sep 88	377	30 May 89
P18L00702	LOADER-TRANSPORTER,	8 Sep 88	377	7 Mar 89
P18L00731	TUNNEL SECTION,GUID	20 Sep 88	365	
P18L00742	ENGINE TURBOJET	27 Sep 88	358	13 Mar 89
P18L00773	AMPLIFIER,AUDIO-RAD	13 Oct 88	342	30 May 89
P18L00811	SHELL,OUTER,BODY	27 Oct 88	328	24 May 89
P18L00844	AMPLIFIER,AUDIO-RAD	8 Nov 88	316	28 Mar 89
P18L00853	TRANSDUCER,PRESSURE	9 Nov 88	315	19 Jun 89
P18L00880	CELL,OPTICAL ELEMEN	23 Nov 88	301	19 Jan 89
P18L00916	CIRCUIT CARD ASSEMB	16 Dec 88	278	21 Dec 88
P18L00920	TAILCONE ASSEMBLY	21 Dec 88	273	23 Feb 89
P19L00041	CABLE ASSEMBLY,POWE	17 Jan 89	246	
P19L00050	WING ASSEMBLY,GUIDE	23 Jan 89	240	5 Jun 89
P19L00057	AMPLIFIER,AUDIO-RAD	24 Jan 89	239	4 Apr 89
P19L00070	RADAR SET,SEMITRAIL	30 Jan 89	233	
P19L00118	GUIDED MISSILE,INTE	21 Feb 89	211	
P19L00122	ANTENNA	23 Feb 89	209	
P19L00126	WING ASSEMBLY,GUIDE	23 Feb 89	209	5 Jun 89
P19L00164	LEVER LOCK RELEASE	8 Mar 89	196	
P19L00169	CIRCUIT CARD ASSY	13 Mar 89	191	
P19L00183	CABLE ASSEMBLY,POWE	15 Mar 89	189	11 Apr 89
P19L00192	TAILCONE ASSEMBLY	17 Mar 89	187	9 Jun 89
P19L00200	ADAPTER, WING	22 Mar 89	182	
P19L00209	GYRO ACCELEROMETER	22 Mar 89	182	
P19L00210	FAN,VANEAXIAL	22 Mar 89	182	15 May 89
P19L00242	POWER SUPPLY ASSEMB	31 Mar 89	173	17 Apr 89
P19L00266	GUIDANCE SHELTER	6 Apr 89	167	15 May 89

CONTINUED ON NEXT PAGE

Continuation of QDR Listing

Action Officer Code: AO3

DOCUMENT MANAGEMENT NR	ITEM NAME	DATE DOCUMENT RECEIVED	CURRENT AGE (DAYS)	DATE LAST REPLY SENT
P19L00313	CABLE ASSY SPEC	13 Apr 89	160	25 Apr 89
P19L00337	VERTICAL GYROSCOPE	22 Apr 89	151	20 Jun 89
P19L00341	STOWAGE BOX	22 Apr 89	151	
P19L00344	TRANSDUCER,PRESSURE	22 Apr 89	151	
P19L00345	TRANSDUCER,MOTIONAL	22 Apr 89	151	
P19L00360	GEAR SPUR	27 Apr 89	146	
P19L00375	CONNECTION BOX	3 May 89	140	3 Jul 89
P19L00396	COUNTER,ELECTRICAL	10 May 89	133	
P19L00399	SHELL,OUTER,BODY SE	10 May 89	133	3 Jul 89
P19L00403	LEM STRUCTURE ASSY	12 May 89	131	
P19L00407	CABLE ASSEMBLY,SPEC	12 May 89	131	25 May 89
P19L00420	CLAMP, HOSE	17 May 89	126	
P19L00421	TARGET DRONE	17 May 89	126	
P19L00439	BOX ASSEMBLY ENGINE	5 Jun 89	107	
PH9L00021	CIRCUIT CARD ASSEMB	21 Apr 89	152	30 May 89

TOTAL NUMBER OF DOCUMENTS = 47

Action Officer Code: AO4

DOCUMENT MANAGEMENT NR	ITEM NAME	DATE DOCUMENT RECEIVED	CURRENT AGE (DAYS)	DATE LAST REPLY SENT
P17L00681	WIRING HARNESS,BRAN	28 Apr 87	876	

CONTINUED ON NEXT PAGE

EXAMPLE OF MONTHLY NOTICE TO CHIEF OF CUSTOMER FEEDBACK OFFICE

Memo to: Chief, Customer Feedback Office 21 Sep 89

From: DRS Database Administrator

Subject: QDRs Overdue for Reply to QDR Originator

According to the current contents of the DRS database on the Unisys 5000 computer, the following documents now require an interim or final reply to the QDR originator. If this listing is incorrect, please advise so that the database can be updated.

DOCUMENT MANAGEMENT NR	ITEM NAME	DATE DOCUMENT RECEIVED	CURRENT AGE (DAYS)	DATE LAST REPLY SENT
P17L00681	WIRING HARNESS,BRAN	28 Apr 87	876	
P17L00725	ROCKET POD,298 MILL	6 May 87	868	5 Dec 88
P17L00730	NETWORK,BETA DETECT	11 May 87	863	
P17L00921	RATE GYRO UNIT	8 Jul 87	805	
P17L01152	CONTROL BOX,GENERAT	10 Sep 87	741	26 Aug 88
P17L01159	WIRING HARNESS,BRAN	14 Sep 87	737	
P17L01292	GUIDED MISSILE,SURF	20 Oct 87	701	16 Feb 88
P17L01368	CIRCUIT CARD ASSEMB	13 Nov 87	677	
P17L01377	CRYOSTAT ASSEMBLY,	19 Nov 87	671	
P17L01378	CRYOSTAT ASSEMBLY,	19 Nov 87	671	
P17L01380	FIN,GUIDED MISSILE	23 Nov 87	667	22 Jul 88
P17L01382	SIGHT,REFLEX	23 Nov 87	667	
P17L01436	PREAMPLIFIER ASSEMB	18 Dec 87	642	21 Mar 89
P17L01442	SIGHT,REFLEX	18 Dec 87	642	
P17L01447	INTEGRATED SIGHT UN	21 Dec 87	639	
P17L01448	INTEGRATED SIGHT AS	21 Dec 87	639	
P17L01451	GUIDANCE SECTION,GU	21 Dec 87	639	
P17L01469	RATE GYRO UNIT	28 Dec 87	632	
P18L00034	CIRCUIT CARD ASSEMB	20 Jan 88	609	17 Nov 88
P18L00059	CIRCUIT CARD ASSEMB	29 Jan 88	600	17 Nov 88
P18L00030	GUIDED MISSILE,SURF	20 Jan 88	609	8 Mar 88

CONTINUED ON NEXT PAGE

Continuation of QDR Listing

DOCUMENT MANAGEMENT NR	ITEM NAME	DATE DOCUMENT RECEIVED	CURRENT AGE (DAYS)	DATE LAST REPLY SENT
P18L00128	ROCKET POD,298 MILL	17 Feb 88	581	5 Dec 88
P18L00174	ELECTRONICS UNIT	1 Mar 88	568	28 Apr 89
P18L00179	CONTROL UNIT	3 Mar 88	566	13 May 88
P18L00193	ROCKET POD,298 MILL	7 Mar 88	562	5 Dec 88
P18L00207	GUIDED MISSILE,INTE	11 Mar 88	558	17 Mar 89
P18L00209	GUIDANCE SECTION,GU	15 Mar 88	554	
P18L00210	GUIDANCE SECTION,GU	15 Mar 88	554	
P18L00211	GUIDANCE SECTION,GU	15 Mar 88	554	
P18L00230	DETECTOR ASSEMBLY,N	21 Mar 88	548	22 Mar 88
P18L00247	MOTOR TACHOMETER AS	25 Mar 88	544	19 May 88
P18L00248	HOUSING ASSEMBLY	25 Mar 88	544	25 Jul 88
P18L00249	HOUSING ASSEMBLY	25 Mar 88	544	
P18L00274	AZ SERVO	31 Mar 88	538	
P18L00275	CONTROL ASSEMBLY	31 Mar 88	538	
P18L00279	CONTROL BOX,GENERAT	31 Mar 88	538	
P18L00280	RECEIVER UNIT,CATS	31 Mar 88	538	
P18L00293	ROCKET POD,298 MILL	6 Apr 88	532	5 Dec 88
P18L00335	MODIFICATION KIT	21 Apr 88	517	
P18L00356	GUIDED MISSILE,INTE	26 Apr 88	512	17 Mar 89
P18L00357	ROCKET POD,298 MILL	26 Apr 88	512	5 Dec 88
P18L00360	VIDEO DISPLAY PANEL	2 May 88	506	
P18L00367	SIGHT UNIT STABLIZE	2 May 88	506	
P18L00387	ROCKET POD,298 MILL	6 May 88	502	
P18L00402	HOUSING ASSEMBLY	12 May 88	496	
P18L00405	GUIDED MISSILE	12 May 88	496	17 Mar 89
P18L00408	ROCKET POD,298 MILL	12 May 88	496	5 Dec 88
P18L00409	ROCKET POD,298 MILL	12 May 88	496	5 Dec 88
P18L00416	SWIVEL ASSEMBLY HYD	16 May 88	492	
P18L00438	ROCKET POD,298 MILL	24 May 88	484	5 Dec 88
P18L00442	HOUSING ASSEMBLY	24 May 88	484	25 Aug 88
P18L00445	SHIPPING AND STORAG	24 May 88	484	
P18L00454	RELAY COVER	1 Jun 88	476	
P18L00487	MISSILE SIMULATION	14 Jun 88	463	
P18L00493	CONSOLE,INSTRUCTOR	15 Jun 88	462	18 Jul 88

CONTINUED ON NEXT PAGE